VŠB – Technická univerzita Ostrava

Fakulta elektrotechniky a informatiky

Katedra informatiky

Realizace síťových služeb na bázi RouterOS x86

Implementation of Network-based Services on RouterOS x86

2013 **Tomáš Pijáček**

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Tomáš Pijáček**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Realizace síťových služeb na bázi RouterOS x86
Implementation of Network-based Services on RouterOS x86

Zásady pro vypracování:

Cílem práce je zrealizovat hraniční směrovač na bázi systému RouterOS x86 s funkcemi firewall, mail server a http server.

1. Zmapujte systém RouterOS x86 a popište na příkladech obecnou metodiku zprovoznění služeb, které nejsou implementovány v RouterOS.
2. Navrhněte vlastní řešení hraničního směrovače na bázi RouterOS x86.
3. Na tomto směrovači navrhněte a zrealizujte funkci stavový firewall, mail server (podpora SMTP a POP3/IMAP) a HTTP server.
4.Stanovte metodiku testování pro srovnání řešení založených na nativním použití služby, použití RouterOS a na HW řešení.
5. Otestujte a srovnejte toto řešení oproti dostupnému HW řešení (Cisco ASA, Cisco router).

Seznam doporučené odborné literatury:

TANENBAUM, Andrew. Computer Networks. 4 edition. England : Prentice Hall, 2002. 912 s. ISBN 978-0130661029.
http://www.mikrotik.com/documentation.html

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Libor Michalek, Ph.D.**

Datum zadání: 18.11.2011
Datum odevzdání: 07.05.2013

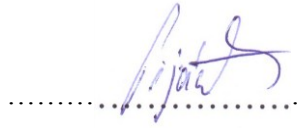
doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 31. července 2013



Bc. Tomáš Pijáček

Poděkování

Rád bych na tomto místě poděkoval vedoucímu diplomové práce panu Ing. Liboru Michalkovi, Ph.D. za vedení a rady při tvorbě této diplomové práce.

Dále bych rád poděkoval panu Ing. Danielu Stříbnému, který mě v psaní této práce velmi podporoval a umožnil mi přístup do školní laboratoře Cisco.

Abstrakt

Cílem této diplomové práce je realizace hraničního směrovače na bázi operačního systému RouterOS x86 s funkcemi firewall, mail server a http server. V první části práce je provedena celková analýza operačního systému RouterOS pro platformu x86. V dalších částech práce se zabývám vlastním návrhem hraničního směrovače a rovněž zde popisuji pracovní postupy, pomocí kterých je návrh realizován. V posledních částech této diplomové práce byla stanovena metodika pro srovnání řešení založených na nativním použití služby, použití RouterOS a na hardwarovém řešení, v konečné fázi byl tento systém porovnán také s konkurenčními řešeními, jako je Cisco ASA a Cisco router.

Klíčová slova

RouterOS, Mikrotik, Cisco, Kvm, Linux, Ubuntu, Apache, Smtp, Pop3, Imap, Postfix, Dovecot, RFC

Abstrakt

The aim of this thesis is the implementation of the border router based on the operating system RouterOS x86 with features firewall, mail server and http server. The first part is a comprehensive analysis of RouterOS operating system for the x86 platform. In the following sections deal with their own design border router and also here with the working procedures by which the proposal is implemented. In the last part of this thesis was established methodology for comparing solutions based on native use of the service, and the RouterOS using a hardware solution, in the final phase of the system was also compared to competing solutions, such as Cisco ASA and Cisco router.

Keywords

RouterOS, Mikrotik, Cisco, Kvm, Linux, Ubuntu, Apache, Smtp, Pop3, Imap, Postfix, Dovecot, RFC

Obsah

Úvod

Velkým fenoménem dneška jsou virtualizace. Tato metoda je známa již řadu let, avšak až v posledních letech její nasazení nabralo na intenzitě, především díky výkonnějšímu a cenově dostupnějšímu hardwaru. Virtualizace oproti klasickému řešení přináší obrovské finanční úspory, jelikož lze provozovat více oddělených serverů s vlastním operačním systémem na jednom fyzickém hardware nebo virtualizovat pouze softwarové aplikace. Díky nasazení virtualizace se rovněž uspoří značná část financí za využívání elektrické energie.

I přes množství kladů, které virtualizace nabízí, je ale před každým jejím nasazením nutné si uvědomit, zda je toto nasazení vhodné (např. provozování virtualizace na směrovači může mít nepříznivý vliv na přenosovou rychlost).

Tato diplomová práce se zabývá návrhem a realizací hraničního směrovače na bázi operačního systému RouterOS pro hardwarovou platformu x86, který je vyvíjen lotyšskou firmou Mikrotik. Jedná se o speciální operační systém, jehož kořeny tkví v linuxových základech a který je přímo určen pro aplikaci v počítačových sítích. Firma Mikrotik v tomto operačním systému implementovala virtualizační nástroje, díky nimž je možné vytvářet virtuální stroje s vlastním operačním systémem.

Diplomová práce je rozdělena do pěti kapitol.

První a druhá kapitola je zaměřena spíše na teoretický výklad - v první kapitole analyzuji operační systém RouterOS, ve druhé kapitole se pak zabývám vlastním návrhem hraničního směrovače.

Třetí kapitola je zaměřena na praktické využití. Jsou zde uvedeny konfigurace a postupy k nasazení služeb, jako jsou stavový firewall, http server, mail server s podporou protokolů SMTP a POP3/IMAP.

Čtvrtá kapitola diplomové práce se zabývá stanovením metodiky pro srovnání řešení založených na nativním použití služby, použití RouterOS a na HW řešení. Dále se v ní věnuji testování a srovnání dosažených výsledků platformy RouterOS x86 oproti dostupným aktivním prvkům Cisco router a Cisco ASA.

Pátá kapitola se věnuje celkovému zhodnocení výsledků této diplomové práce a glosuje možnosti využití virtualizace na hraničním směrovači.

1 Operační systém RouterOS

1.1 Co je to RouterOS

RouterOS (Router Operating System, dále jen ROS) je komerční operační systém založený na bázi operačního systému Linux v2.6 kernel. Byl vyvinut lotyšskou firmou MikroTik založenou v roce 1995. Obsahuje nepřeberné množství síťových funkcí, a je tak primárně určený pro poskytovatele internetového připojení (ISP). ROS je distribuován v podobě instalačních balíčků .npk nebo klasického ISO souboru. Mezi nejpoužívanější funkce ROS patří hlavně firewall, omezování šířky pásma, směrování, nasazování do bezdrátových sítí jako přístupový či klientský bod.

ROS je provozován jako výchozí operační systém na zařízeních typu RouterBoard a také je možné jej instalovat na hardwarové platformy x86, mips, i386 a powerpc. U RouterBoardů jsou licenční úrovně dodávány s již předinstalovaným ROS, zatímco hardwarové platformy je nutné jednu z licenčních úrovní zakoupit. [1]

1.2 Licence

ROS je v současné době dostupný v šesti licenčních úrovních.

Jedinou volně dostupnou licenční verzí je verze L0 určená především pro testování a ukázku veškerých dostupných funkcí. Po nainstalování je k dispozici 24 hodin, poté přestane být funkční. Zdarma je rovněž licenční verze L1, která je ale pouze jakousi demo verzí s velmi omezenými funkcemi. Vyžaduje registraci a zadání licenčního kódu, který je uživateli zaslán na e-mail. Existovala také licenční verze L2, která již v současné době není k dispozici.

Zakoupit je možno licenční úrovně L3, L4, L5 a L6. Po zakoupení licenční úrovně je po uživateli opět vyžadováno zadání licenčního klíče zaslaného na e-mail. Tento licenční klíč představuje blok symbolů, které je možné zadat buď přímo v terminálu, v prostředí winbox, nebo webovém rozhraní webfig. Úroveň L3 je dostupná pouze při odběru minimálně 100 kusů (existují však i RouterBoardy, u nichž je tato úroveň již předinstalovaná) a využívá se hlavně v bezdrátových sítích v módu klient. Nejrozšířenější úrovní je úroveň L4. Je to dáno především dostatečným množstvím funkcí a příznivou pořizovací cenou, která se v současnosti pohybuje okolo 700 Kč. Licenční úroveň L5 je ve svých funkcích limitována pouze minimálně a lze ji pořídit za cenu kolem 1200 Kč. Licenční úroveň L6 je nejvyšší úrovní a za cenu až 3300 Kč je funkčně plně neomezena. [2,3]

Omezení, kterými se výše specifikované úrovně navzájem liší, jsou blíže popsána v tabulce č. 1.

Tabulka 1 - přehled licenčních úrovní RouterOS [2]

Level number	0	1	3	4	5	6
Initial Config Support	-	-	-	15 days	30 days	30 days
Wireless AP	24h trial	-	-	yes	yes	yes
Wireless Client and Bridge	24h trial	-	yes	yes	yes	yes
RIP, OSPF, BGP protocols	24h trial	-	yes(*)	yes	yes	yes
EoIP tunnels	24h trial	1	unlimited	unlimited	unlimited	unlimited
PPPoE tunnels	24h trial	1	200	200	500	unlimited
PPTP tunnels	24h trial	1	200	200	500	unlimited
L2TP tunnels	24h trial	1	200	200	500	unlimited
OVPN tunnels	24h trial	1	200	200	unlimited	unlimited
VLAN interfaces	24h trial	1	unlimited	unlimited	unlimited	unlimited
HotSpot active users	24h trial	1	1	200	500	unlimited
RADIUS client	24h trial	-	yes	yes	yes	yes
Queues	24h trial	1	unlimited	unlimited	unlimited	unlimited
Web proxy	24h trial	-	yes	yes	yes	yes
User manager active sessions	24h trial	1	10	20	50	Unlimited
Number of KVM guests	none	1	Unlimited	Unlimited	Unlimited	Unlimited

1.3 RouterOS balíčky

ROS podporuje mnoho různých funkcí a díky tzv. „balíčkovacímu“ systému je možné zvolit si pouze funkce, které chce uživatel využívat. Popis dostupných jednotlivých balíčků je znázorněn v tabulce č. 2.

Tabulka 2 - přehled dostupných balíčků [2]

Balíček	**Popis funkce**
system	Nejnutnější balíček, bez kterého by nebylo možno provozovat ROS
ppp	Podpora tunelů PPP, PPTP, L2TP, PPPoE
dhcp	Podpora DHCP client, server
advance-tools	Pokročilé nástroje Netwatch, Pingers
calea	Nástroj pro sběr dat
gps	Podpora pro GPS zařízení
hotspot	Podpora pro vytvoření Hotspotu pro uživatele
ipv6	Podpora protokolu IPv6
isdn	Podpora ISDN

kvm	Podpora virtualizace KVM
lcd	Podpora pro LCD display
mpls	Podpora protokolu Mpls
multicast	Podpora šíření multicastů
ntp	Podpora Ntp server, client
routerboard	Podpora pro RouterBoard
routing	Podpora dynamických protokolů
security	Podpora zabezpečení, IPsec, SSH
ups	Podpora pro UPS
user-manager	Podpora Radius serveru a web serveru
wireless	Podpora pro bezdrátové karty

1.4 Požadavky na hardware

Jak již bylo řečeno, ROS je možné provozovat na různých platformách architektury (x86, mips, i386, powerpc) a rovněž na RouterBoardech, na kterých se momentálně vyskytuje nejčastěji. Těch je v současnosti velká škála. RouterBoardy se liší jak svou výkonností, tak kapacitou flash paměti, operační pamětí, počtem miniPCI slotů, ethernet portů atd. Jejich hlavní potenciál tkví v jejich velikosti a hlavně nízké spotřebě energie.

Více je ale potenciál ROS využit při použití na platformě x86, jelikož je její výkon narozdíl od výkonu RouterBoardů srovnatelný s profesionálními směrovači řady Cisco, další výhodou provozu ROS na platformě x86 je také široká modulace osazení síťovými kartami. Tato varianta je ale oproti provozu na RouterBoardech nákladnější, poněvadž PC má větší rozměry a spotřebuje daleko více elektrické energie.

ROS podporuje jak multi-core, tak multi-CPU počítače, nejnovější ethernet karty s přenosnou rychlostí 10 Gigabitů, 802.11a/b/g/n bezdrátové karty a 3G modemy.

Pro instalaci ROS je možné použít zaváděcí médium IDE/SATA nebo USB zařízení a pro možné nainstalování operačního systému pevný disk (HDD), copmactflash (CF), paměťovou kartu (SD) či solid state drive (SSD) o velikosti minimálně 64MB. Minimální velikost operační paměti musí být nejméně 32MB a frekvence procesoru by měla dosahovat 100Mhz. [2]

1.5 Nástroje pro správu

Operační systém ROS je možné konfigurovat různými metodami, od místního přístupu s klávesnicí a monitorem, sériovou linku s terminálovou aplikací až po vzdálený přístup, jako je telnet či SSH. Existují celkem 3 základní způsoby managementu ROS – WinBox, WebFig, konzolové připojení.

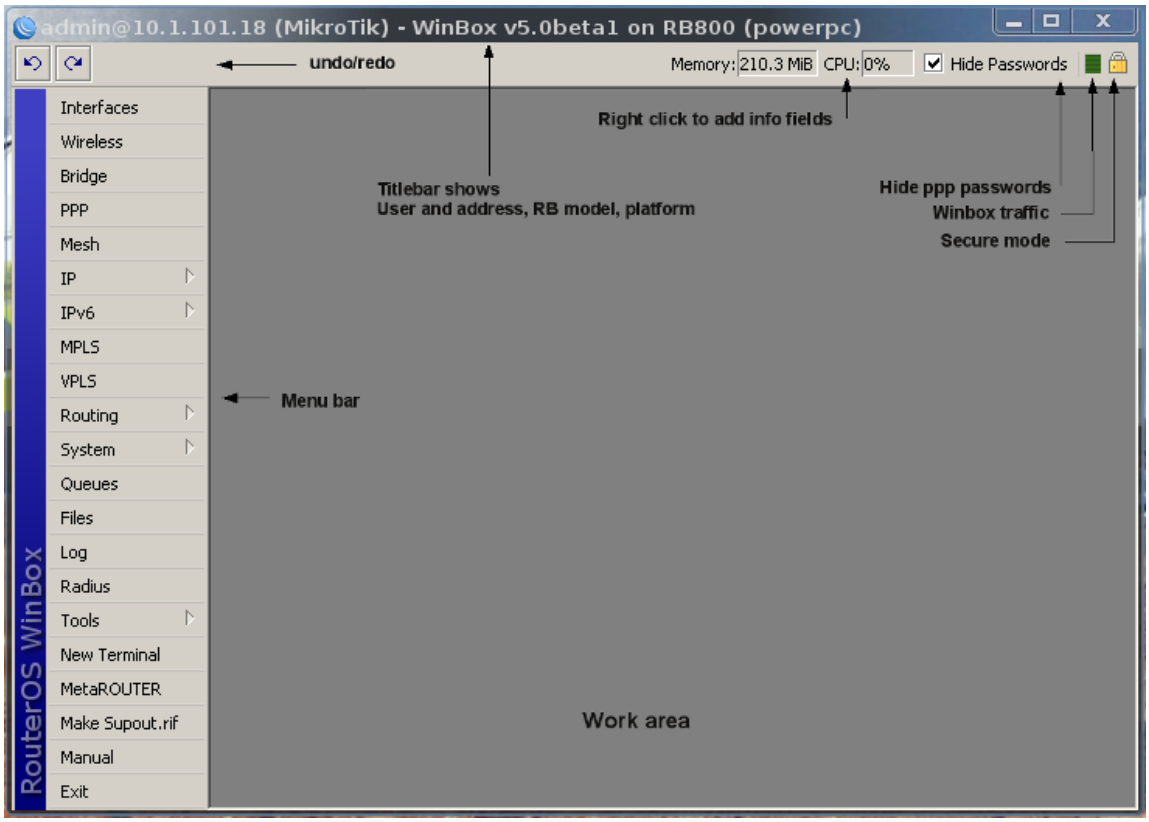
Čtvrtou, rozšířenou, možností je vytvoření vlastní aplikace díky programovému rozhraní API (Application programming interface).

1.5.1 WinBox

Jedná se o jednoduchý a rychlý nástroj určený pro správu ROS, který není nutné instalovat, a proto ho můžeme lehce přenášet na jakémkoli úložném médiu. WinBox je naprogramován v nativním Win32 kódu a je ho možné spustit i na operačních systémech Linux či Mac OSX pomocí utility Wine. Tento nástroj je pro management mezi uživateli bezpochyby nejoblíbenější. Po jeho spuštění se díky technologiím MNDP (Mikrotik Neighbor Discovery Protocol) a CDP (Cisco Discovery Protocol) automaticky detekují připojitelná zařízení, včetně těch, která nejsou s WinBoxem kompaktibilní (např. směrovače Cisco atd.). K zařízení je možné se připojit jak pomocí IP adresy, tak pomocí MAC adresy. Připojení pomocí MAC adresy se ale moc nedoporučuje, jelikož není 100% spolehlivé z důvodu využívání broadcastového vysílání. WinBox rozhraní bylo navrženo především pro intuitivní ovládání. [2]

Rozhraní se skládá z následujících tří částí:

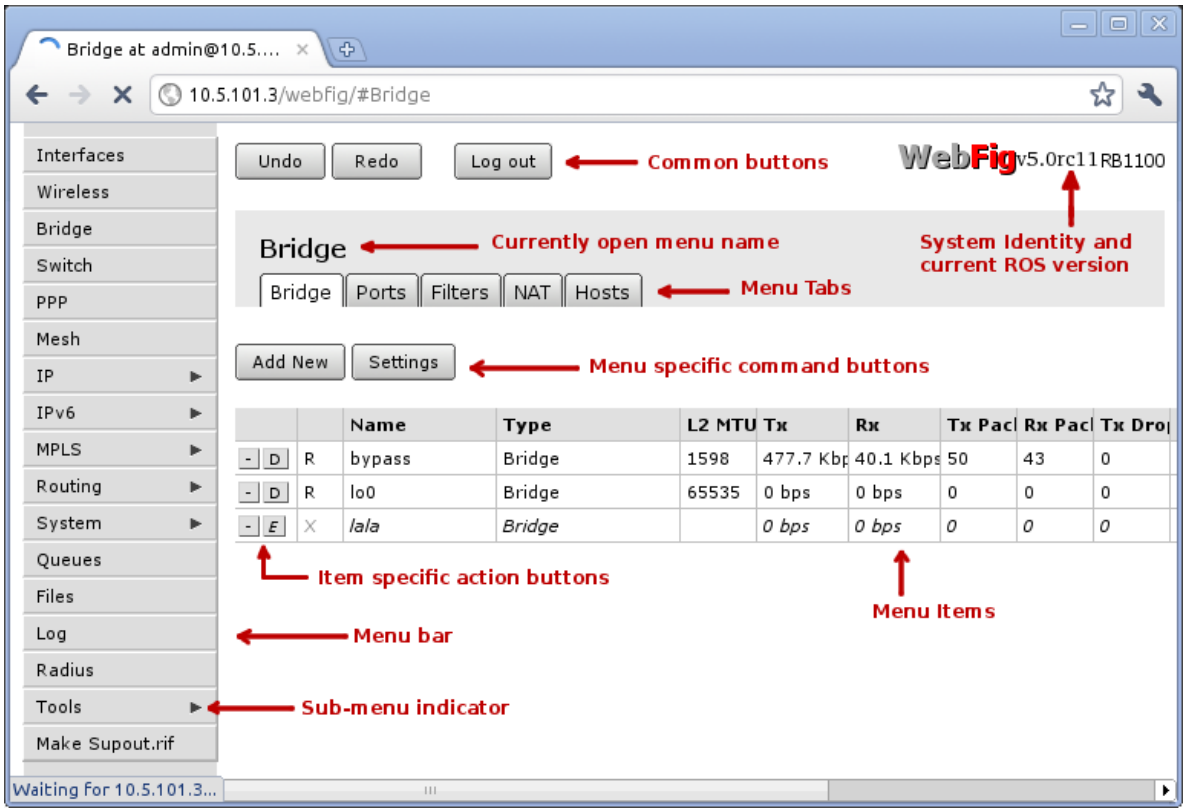
- Hlavní panel nástrojů - jsou zde zobrazeny informace, jako jsou využití CPU a paměti
- Menu bar – seznam všech dostupných funkcí v závislosti na nainstalovaných balíčcích
- Pracovní plocha – plocha, kde jsou otevřeny všechny nabídky



Obrázek 1 - prostředí WinBox [2]

1.5.2 WebFig

Další možností, jak spravovat ROS, je webové rozhraní zvané WebFig, které od 5. verze ROS nahradilo dříve používaný WebBox. Pro přístup k této utilitě není nutný žádný software kromě webového prohlížeče s podporou JavaScriptu, což dělá tento nástroj nezávislým na použité platformě, a může tak být použit ke konfiguraci směrovače přímo z různých mobilních zařízení. WebFig je koncipován jako alternativa WinBoxu, oba mají podobné rozvržení a přístup téměř do všech funkcí ROS [2].



Obrázek 2 - prostředí WebFig [2]

1.5.3 Konzolové připojení

Konfigurace ROS je také možná přes konzoli pomocí textových terminálů, a to buď pomocí sériového portu, Telnetu, nebo šifrovaného přenosu SSH. Konzole umožňuje konfiguraci směrovače pomocí textových příkazů, které můžeme snadno vyvolat nápovědou při stisku klávesy „?“. Často používanými příkazy jsou „print“ (používá se pro výpis) a „set“ (pro nastavení vlastností). Pro usnadnění práce se využívá klávesa „tab“, která dokončí rozepsaný příkaz (tuto funkci využívá i Cisco). [2]

Obrázek 3 - prostředí konzole

1.6 Funkce operačního systému RouterOS

Operační systém ROS má nepřeberné množství funkcí, pro potřeby této práce jsem tedy vybral ty funkce, které byly v praktické části využity.

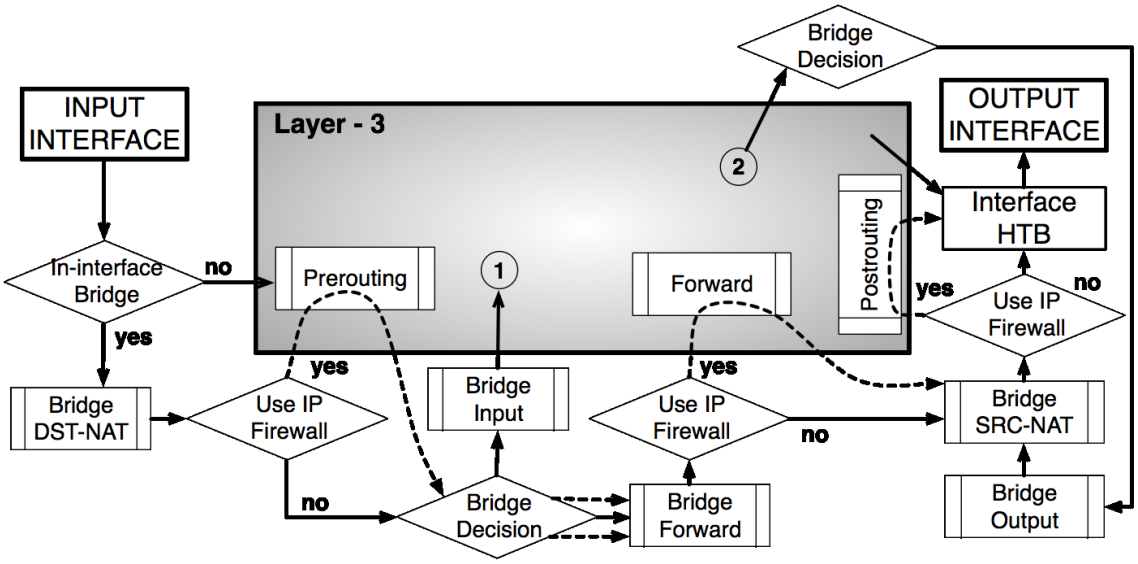
1.6.1 Směrování

Směrování je určeno k určování cest paketů v TCP/IP sítích. Může být statické, nebo dynamické, ROS podporuje celou řadu směrovacích protokolů. [2]

- v případě IPv4 podporuje RIPv1, RIPv2, OSPFv2, BGPv4
- v případě IPv6 podporuje RIPng, OSPFv3 a BGP.

1.6.2 Firewall

V současné době je ochrana dat a zabezpečení hlavní prioritou při budování počítačových sítí. Firewall si můžeme představit jako síťový prvek, díky němuž můžeme zabezpečit či řídit provoz počítačové sítě. Tento prvek se umisťuje na hranicích sítě, aby zabezpečil síť proti vnějšímu ohrožení. Firewally nepředstavují pouze jedinou možnost ochrany, ale díky jejich využití lze snížit riziko napadení sítě. V ROS se Firewall velmi podobá linuxovému nástroji IPtables. Implementovaný Firewall v ROS se skládá z několika tabulek, přičemž každá tabulka je určena k různému druhu práce s pakety. Každá tabulka pak obsahuje vlastní řetězce, které dále definují pravidla, jak bude s paketem naloženo (podrobnější popis těchto tabulek je vysvětlen níže). Pro pochopení, jak firewall v ROS pracuje, je dobré znát tok paketů směrovačem (tento tok je znázorněn na obrázku číslo 4). Každý příchozí či odchozí paket postupně prochází minimálně jedním řetězcem, kde je testován, zda vyhovuje definované podmínce. [2]



Obrázek 4 - Průtok paketu směrovačem [2]

1.6.2.1 Filter

Jedná se o tabulku, která je určená pro filtrování paketů. Tato tabulka je dále rozdělena do tří základních řetězců, a to:

- Input – pravidla aplikující se na pakety, které přichází některým rozhraním a končí na směrovači,
- Forward – pravidla pro pakety, které směrovačem pouze prochází,
- Output – pravidla pro pakety, které vznikly na směrovači a odcházejí některým rozhraním.

V každém z těchto řetězů je možné definovat následující akce:

- Accept – slouží pro přijetí paketu,
- Drop – slouží pro zahození paketu,
- Reject – slouží pro odmítnutí paketu,
- Další akce typu Jump, Log, Return apod.

1.6.2.2 Nat

Pomocí tabulky Nat můžeme vytvářet překlady adres do sítě nebo ze sítě, dále můžeme přesměrovávat porty, nebo dokonce i celé rozsahy IP adres. Tabulka Nat obsahuje dva řetězce typu:

- Srcnat – slouží pro překlad zdrojových adres,
- Dstnat – slouží pro překlad cílových adres.

V každém z těchto řetězů je možné definovat následující akce:

- Masquerade – slouží pro modifikaci zdrojového paketu na adresu síťového rozhraní, kterým opustil firewall,
- Dst-Nat – slouží pro nahrazení cílové adresy / portu na jinou specifickou adresu / port,
- Src-NAT - slouží pro nahrazení zdrojové adresy / portu na jinou specifickou adresu / port,
- Další akce typu Accept, Jump, Netmap apod.

1.6.2.3 Mangle

Tabulka Mangle slouží k modifikaci a značkování paketů. Označené pakety jsou dostupné pouze v rámci směrovače a nepřenáší se dále do sítě. Tato tabulka obsahuje řetězce:

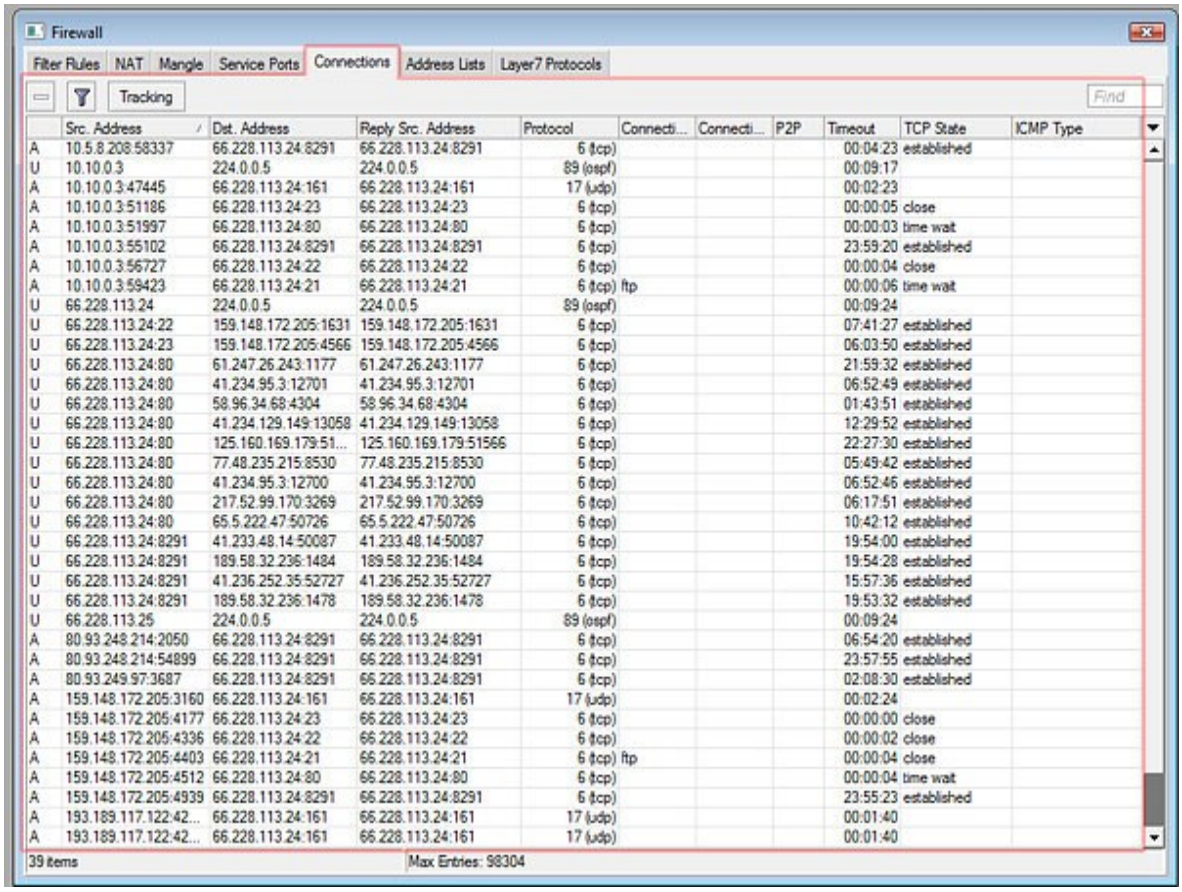
- Prerouting – slouží pro změnu cíle, pravidlo se aplikuje ještě před průchodem do firewallu,
- Postrouting – aplikuje se až po průchodu firewallem, slouží pro změnu zdroje,
- Input, Forward, Output.

1.6.2.4 Connections tracking

Součásti ROS je i stavový firewall, který pracuje na transportní vrstvě referenčního modelu ISO/OSI. Jedná se o typ firewallu, který je schopen sledovat a udržovat všechny navázané TCP/UDP relace, nacházející se v Connections tracking (viz znázornění na obrázku číslo 5). Díky udržovaným záznamům o spojení se tak nemusí prohledávat celý seznam definovaných bezpečnostních pravidel, jelikož pakety jsou povolovány na základě uloženého záznamu v paměti. [2]

Stavový firewall rozlišuje několik stavů, a to:

- New – základní stav začínající nové spojení,
- Established – paket již patří k nějakému navázanému spojení,
- Related – paket navazuje další nové spojení,
- Invalid – paket není součástí známého spojení.



Obrázek 5 - Connections tracking [2]

1.6.3 Virtualizace

Virtualizace byla od společnosti Mikrotik velkým krokem dopředu. Umožňuje totiž na jednom fyzickém zařízení provozovat několik různých operačních systémů v závislosti na použité architektuře. Díky této možnosti mohou uživatelé ROS provozovat tzv. non-RouterOS software, a rozšířit tak funkce, které nejsou přímo implementovány v ROS (např. různá linuxová distribuce či windows). První podpora virtualizace přišla ve verzi ROSv3.11 a jednalo se o softwarovou podporu Xen určenou pro platformu x86. Počínaje verzí v3.21 vznikla podpora MetaRouteru pro platforu mipsbe a dále ppc. Třetí a zároveň poslední možnost virtualizace přinesla verze v3.26. Jednalo se o hardwarovou podporu Kvm určenou pro platformu x86. [2]

1.6.3.1 Virtualizace Xen

Jedná se o linuxový Open-Source projekt založený na Xen Virtual Machine vyvíjen firmou Citrix. Xen je hypervizor, který poskytuje rozhraní pro virtualizaci hardwaru a běh více operačních systémů, které podporují paravirtualizaci na jedné x86 platformě. Pokud budeme chtít používat virtualizaci Xen, je nutné nainstalovat balíček xen.npk. Tato možnost virtualizace byla ale od verze ROSv4.4 ukončena. [2]

1.6.3.2 MetaRouter

Tato technologie se využívá u platforem mipsbe Routerboardech série RB400, RB700 a ppc RB1000, RB1100, RB1100AH a RB800. Slouží pro běh více operačních systému ROS nebo OpenWRT s omezením až osmi virtuálních strojů (do budoucna se plánuje podpora až na 16 virtuálních strojů). [2]

1.6.3.3 Virtualizace Kvm

Kvm (Kernel-based Virtual Machine) je virtualizační řešení pro platformu x86 vyvinuté firmou Qumranet, kterou převzala společnost Red Hat. Tato technologie podporuje pouze plnou virtualizaci, tudíž je nutné mít k jejímu běhu odpovídající hardware, což znamená vlastnit jeden z procesorů s podporou technologie Intel VT-x nebo AMD-V. Výhodou oproti virtualizaci Xen je fakt, že umožňuje spouštět operační systém bez modifikace jádra.

Ke správě virtualizovaných systémů se využívá buď konzole (správa pomocí příkazů), nebo VNC nástroj, který slouží jako virtuální displej. Každý vytvořený host vyžaduje alespoň 16Mb paměti RAM a dostatek úložného prostoru pro souborový obraz vybraného OS. Abychom mohli využít tento virtualizační nástroj, je nutné mít nainstalovaný balíček kvm.npk, díky němuž se podpora KVM v RouterOS aktivuje. [2]

1.6.4 Podpora dalších služeb

- Podpora tunelů: L2TP, PPTP, PPP, IPSec
- DHCP
- DNS
- NTP
- HTTP proxy
- QoS
- Skriptování
- Scheduler

1.7 Instalace RouterOS

Instalovat ROS lze více způsoby. Zvolená metoda instalace závisí především na tom, jaký typ hardware bude použit. Dosti často se využívá aplikace Netinstall, která umožňuje nainstalovat ROS na PC nebo RouterBoard přes ethernetovou síť. Díky této aplikaci je možné znovu nainstalovat ROS i v případech, kdy selhala předchozí instalace, a na zařízení nebylo možno přistoupit či byla ztracena přístupová hesla. ROS je také možné instalovat přímo na disky, jako jsou USB, CF nebo IDE. Tato diplomová práce se zabývá instalací ROS na platformu x86, tudíž se nabízí využití daleko snazší metody, a sice stáhnout již dopředu připravený ISO obraz ROS z webových stránek výrobce. Ten stačí následně vypálit na CD/DVD médium a při startu PC z něj načíst. Současně je důležité mít na paměti, že při instalaci ROS se smažou všechna data na disku HDD a ROS bude fungovat pouze jako jediný operační systém.

Postup instalace:

1. Stažení ISO obrazu ROS z webových stránek http://www.mikrotik.com/download.html
2. Vypálení ISO souboru na CD/DVD médium
3. Načtení z CD-ROM mechaniky
4. Volba požadovaných balíčků, které chceme instalovat - je možné vybrat všechny balíčky, nebo minimálně jeden (system), který je nutný k provedení instalace. Po stisknutí klávesy „i“ se nás systém dotáže, zda chceme použít staré nastavení, to platí pouze v případě, že máme již nějakou verzi ROS nainstalovanou. Postup instalace je znázorněn na obrázku č. 6
5. Po úspěšné instalaci je možno se do ROS přihlásit. Jako výchozí přihlašovací jméno je „*admin*“. Heslo ROS nevyžaduje.

Obrázek 6 - Proces instalace ROS

2 Návrh hraničního směrovače

V této části diplomové práce popisuji návrh řešení hraničního směrovače na bázi RouterOS x86.

2.1 Použitý hardware směrovače

Hardware, který byl použit v rámci této diplomové práce, plně dostačoval jejím potřebám a je popsaný v níže uvedené tabulce.

Tabulka 3 - použitý hardware směrovače

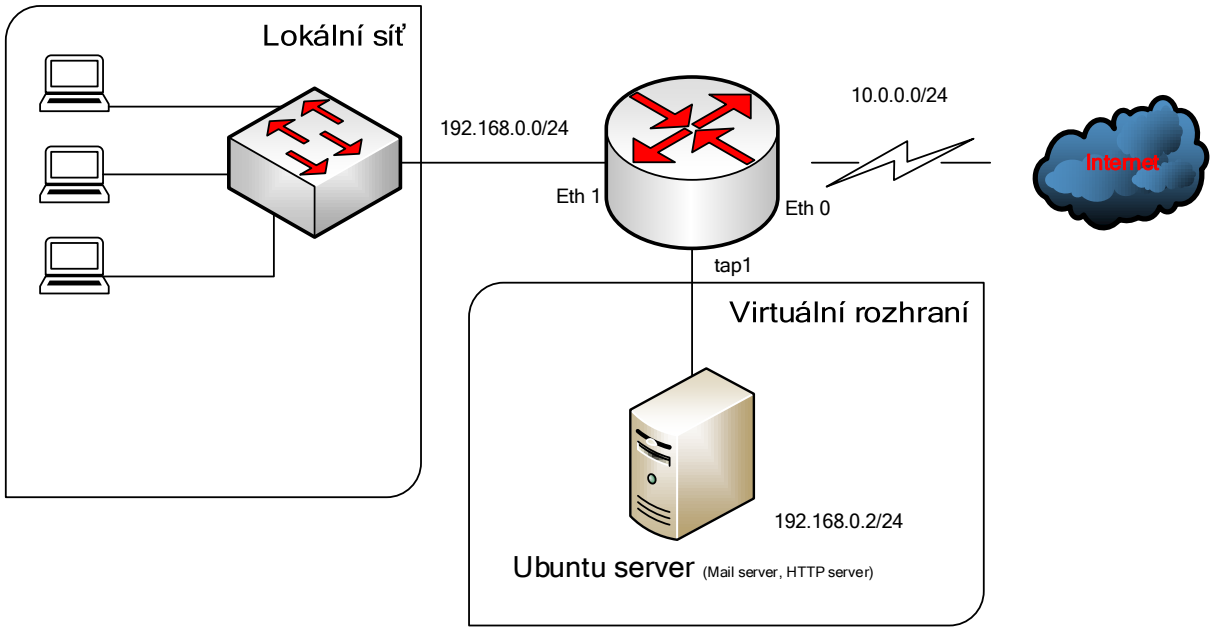
CPU	INTEL CORE 2 DUO PROCESSOR P8400
Čipová sada	Intel P35
Operační paměť	2GB DDR2-800 (400Mhz)
Pevný disk	WDC WD6400, SATA-II
Síťové rozhraní Eth0	Realtek RTL8168B/8111B PCI-E
Síťové rozhraní Eth1	OvisLink LFE-8139ATX Fast Ethernet Adapter
Verze ROS	5.20, licence L4

2.2 Koncepce řešení

Hraniční směrovač je posledním řízeným směrovačem, který stojí před vstupem do Internetu. Tento směrovač řídí provoz vedoucí dovnitř sítě, uvnitř sítě a také provoz vycházející ven ze sítě. Základem tohoto návrhu je realizace hraničního směrovače, který plní funkci stavového firewallu a na kterém jsou díky implementovaným virtualizačním nástrojům provozovány služby SMTP, POP3/IMAP a HTTP.

Z důvodu vyšší bezpečnosti jsou adresy stanic lokální sítě překládány na jednu veřejnou IP adresu, která vstupuje do Internetu.

Síťové rozhraní Eth1 a virtuální rozhraní hostovaného operačního systému jsou spolu připojeny do vytvořeného síťového mostu označeného jako *kvm_ubuntu*. IP adresy byly zvoleny v rozsahu 192.168.0.0/24. Hostovaný operační systém měl napevno definovanou IP adresu 192.168.0.2/24, pro další vstupující zařízení do tohoto segmentu sítě byl použit DHCP server, který automaticky přidělí potřebnou IP adresu. Na obrázku číslo 7 je znázorněna síťová topologie řešení.



Obrázek 7 - topologie zapojení

2.3 Výběr virtualizovaného operačního systému

ROS neumožňuje nativně provozovat služby, jako jsou poštovní či webový server. Abych mohl tyto služby provozovat, je nutné vybrat takový operační systém, který umožňuje jejich realizaci. Volba operačního systému je velmi důležitá například z hlediska ceny či licence. Z tohoto důvodu jsem volil operační systém stavěný jako tzv. open source, což představuje software s otevřeným zdrojovým kódem a neomezenou licencí, který můžeme používat zcela zdarma. Jako příklad operačního systému, který je založený na samotné myšlence open source, je Unixový systém GNU/Linux, který jsem taktéž zvolil pro realizaci již zmiňovaných služeb.

2.3.1 GNU/Linux

GNU/Linux je operační systém, na kterém spolupracují jak jednotlivci, tak komerční společnosti či různé organizace z celého světa. Hlavním důvodem vzniku GNU/Linux je nabídnout potencionálním uživatelům svobodný operační systém bez restrikcí a různých omezení. Tento operační systém je velmi flexibilní a je možné jej provozovat na desktopech, serverech, směrovačích, či dokonce na mobilních telefonech. GNU/Linux nepředstavuje pouze jenom jeden konkrétní systém, ale zahrnuje celou řadu nezávislých projektů, jejichž integrací vzniká konkrétní operační systém. Tento operační systém si takto můžeme vlastnoručně navrhnout, ale toto řešení je značně složité a časově náročné, proto se doporučují různé linuxové distribuce, které nabízejí již sestavené operační systémy založené na GNU/Linux (příkladně Debian, Ubuntu, CentOS, Fedora, atd.). [3]

2.3.2 Ubuntu

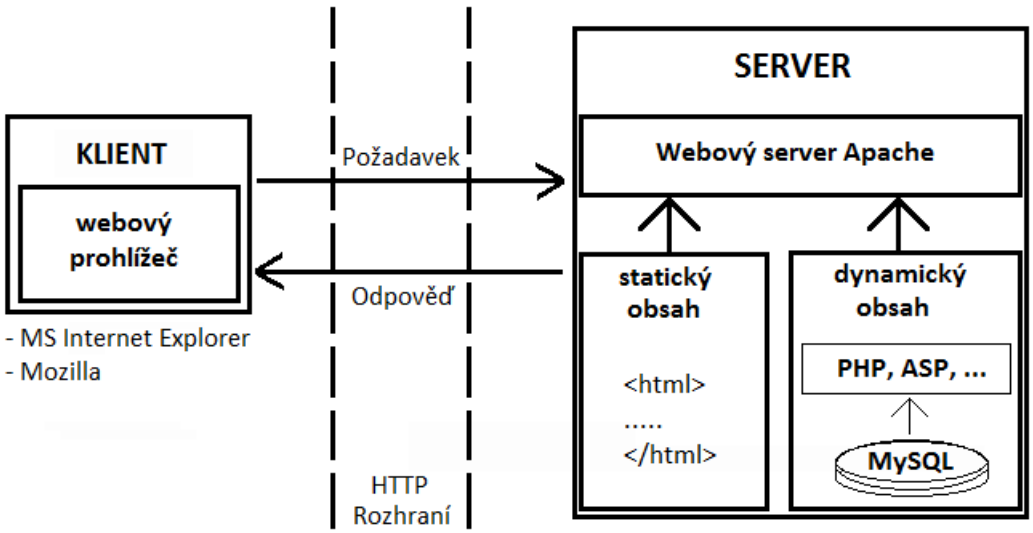
Ubuntu je svobodná linuxová distribuce vhodná pro laptopy, servery nebo stolní počítače. Tato distribuce vychází z distribuce Debian GNU/Linux, která má dlouholetou tradici. Distribuce Ubuntu vznikla v roce 2004 a jejím zakladatelem byl Mark Shuttleworth. První verze systému nesla jméno Ubuntu 4.10 Warty Warthog. Velmi pozitivní vlastností Ubuntu od Debianu jsou pravidelné bezpečnostní aktualizace po dobu minimálně 18 měsíců a vydání nové verze každých 6 měsíců. Jednou za dva roky je vydána tzv. LTS (Long Term Support) verze, s níž získáme 3letou podporu pro osobní počítač a 5letou podporu pro server. Důvodů, proč jsem zvolil právě distribuci Ubuntu, je několik [4] :

- velká podpora Ubuntu ze strany výrobců HW,
- velká multimediální podpora,
- 5letá podpora aktualizací pro server,
- kvalitní základy Debianu,
- jednoduché konfigurace.

2.4 Popis a výběr služeb řešící webový server

2.4.1 Architektura webového serveru

Úkolem webového serveru je zpracovávat http (Hyper Text Transfer Protocol) požadavky od klientů (klientem může být například webový prohlížeč, požadavkem se rozumí odeslání webové stránky, která je nejčastěji ve formátu HTML či jemu podobným formátům např. XHTML, XML). Princip samotné komunikace webového serveru znázorňuje obrázek č. 8. V praxi je nejčastěji webový server tvořen pomocí trojice aplikací: Apache, MySQL a PHP. O této trojici se často hovoří jako o tzv. LAMP (Linux, Apache, MySQL, PHP), který představuje dobře osvědčenou sadu svobodného softwaru určeného jako platformu pro implementaci webových stránek. [5]



Obrázek 8 - Princip webového serveru

2.4.2 Apache

V současné době existuje celá řada webových serverů. Ke své práci jsem si vybral Apache, jelikož se jedná o open source, který je zároveň nejpoužívanějším multiplatformním řešením webového serveru vyvinutý společností ASF (Apache Software Foundation). Apache nabízí velkou škálu tzv. MPM (MultiProcessing modulů), což mu dovoluje se přizpůsobit potřebám systému, na kterém běží. Díky těmto MPM lze Apache nastavit jako čistě vláknový server, procesorově orientovaný server, či jejich možnou kombinací. [5]

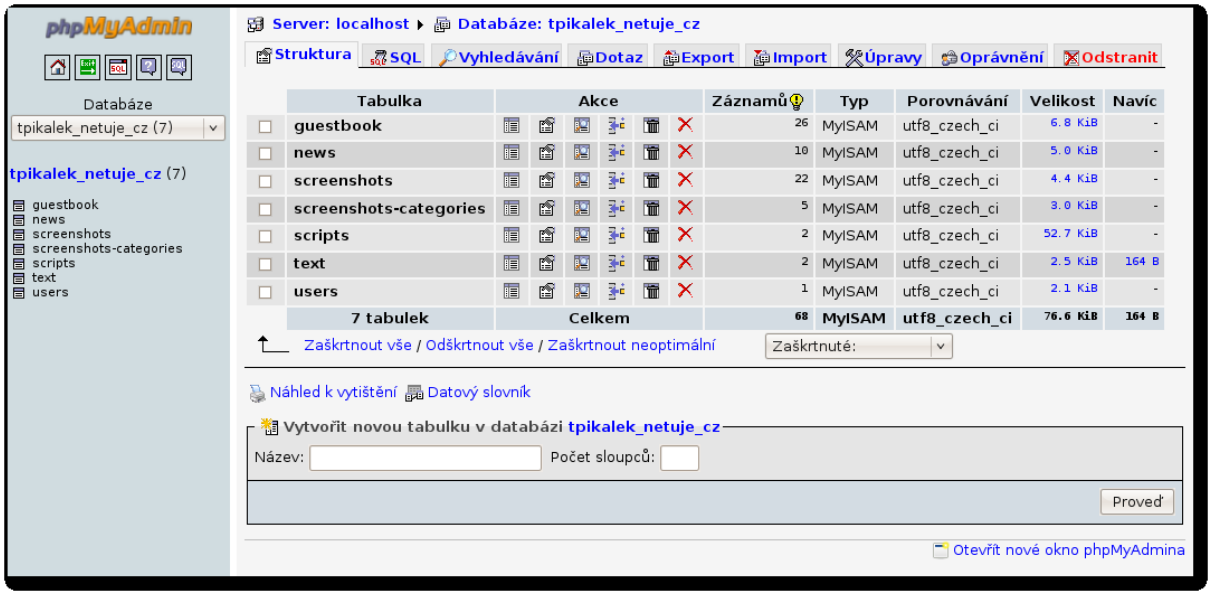
Aktuální verze v době psaní diplomové práce - 2.2.22.

2.4.3 MySQL

Velká většina dnešních síťových aplikací potřebuje ke svému chodu vhodné uložiště dat. Často takovýmto uložištěm bývá MySQL databáze, která je multiplatformní a je poskytována pod dvojím

licencováním (je možno využít bezplatnou licenci nebo komerční placenou). Tato databáze je založena na SQL (Standard Query Language) jazyku, nejčastěji se nasazuje ve spojení Apache a PHP jako základní software pro provoz webového serveru. Pro jednoduchou správu databáze MySQL se nejčastěji používá oblíbená aplikace phpMyAdmin, viz obrázek č. 9. Tento programový nástroj je napsán v jazyce PHP a přistupuje se do něj pomocí webového rozhraní. [6]

Aktuální verze v době psaní diplomové práce - 5.5.20.



Obrázek 9 - Ukázka aplikace phpMyAdmin[8]

2.4.4 PHP

Mnoho dnešních webových serverů podporuje skriptovací jazyk PHP (rekurzivní zkratka PHP: Hypertext Preprocessor), který je díky své jednoduchosti, modularitě a dokumentaci velmi oblíbený. Je určen především pro programování dynamických internetových stránek či webových aplikací. Díky modularitě jej můžeme rozšiřovat o další funkce, jako jsou například podpora konektivity s většinou databázových systémů, generování PDF dokumentů nebo práce se soubory. [7]

Aktuální verze v době psaní diplomové práce - 5.3.10.

2.5 Popis a výběr služeb řešící poštovní server

2.5.1 Architektura elektronické pošty

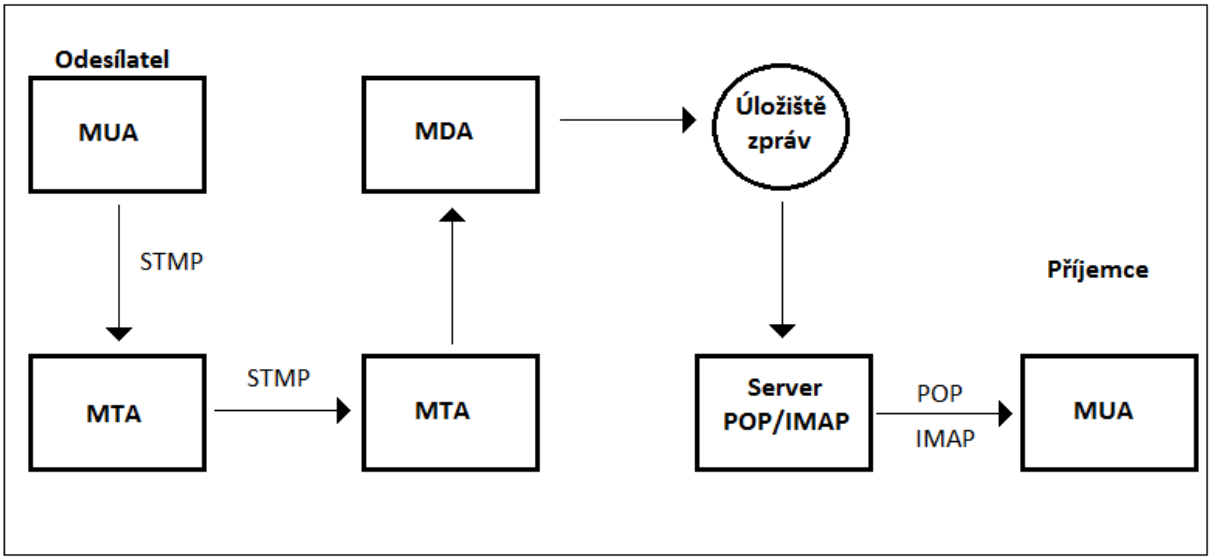
Elektronická pošta je bezpochybně primárně nejpoužívanější prostředek určený k doručování zpráv. Tak jako většina internetových služeb má založenou architekturu na typu klient-server a pro přenos elektronických zpráv mezi stanicemi využívá protokol SMTP. Vyměňování elektronických zpráv je založeno na několika blocích. Těmto blokům říkáme „agenti elektronické pošty“ a dělíme je na [9]:

- **MUA** – Mail User Agent (poštovní uživatelský agent) - klientský software jako například Outlook, Thunderbird a další, určený pro vytváření, odesílání a přijímání elektronických zpráv. Tyto zprávy odesílá pomocí přenosového agenta MTA. K přijímání elektronických zpráv využívá protokol POP3 či IMAP.
- **MTA** – Mail Transfer Agent (poštovní přenosový agent) - server, který slouží k přijímání a doručování zpráv. MTA jsou například Sendmail, Postfix nebo Qmail.
- **MDA** – Mail Delivery Agent (poštovní doručovací agent) - software, který se stará o zpracování a uložení zpráv. MDA zprávy často při doručování filtruje, může také provádět antivirovou nebo antispamovou kontrolu.

Proces, jak celá komunikace mezi agenty elektronické pošty probíhá, zachycuje obrázek č. XX, na kterém můžeme pozorovat jednoduché vyslání poštovní zprávy od odesílatele k příjemci. MUA předá poštovní zprávu pomocí protokolu SMTP poštovnímu serveru, na kterém běží poštovní agent MTA. Agenti MTA mají na starosti veškerou práci s přesunem zpráv z jednoho systému na druhý. Když MTA obdrží požadavek na příjem zpráv elektronické pošty, rozhodne, zda danou zprávu přijme, či nikoliv. Obvykle MTA přijímá zprávy pro své vlastní lokální uživatele, pro jiné systémy, kterým umí zprávu předat, nebo zprávy od uživatelů, systémů či sítí, které mohou elektronickou poštu předávat na jiné cíle. V okamžiku, kdy daný MTA přijme zprávu, se musí rozhodnout, co s ní udělá. Zprávu může odeslat nějakému uživateli na svém systému, nebo ji může předat dalšímu MTA agentovi. Pokud MTA nedokáže zprávu doručit či předat dále, odešle ji zpět původnímu odesílateli, nebo na tuto událost upozorní správce systému. Jakmile dorazí zpráva na MTA, který je konečným cílem, je předána MDA agentovi, který zajistí její konečné doručení. MDA si může zprávu uložit jako prostý soubor, nebo ji může předat databázi elektronické pošty.

Tyto možné metody uložení elektronických zpráv můžeme spojit termínem „úložiště zpráv“. V tomto úložišti jsou umístěny zprávy do té doby, než si je patřičný příjemce vyzvedne. K převzetí zprávy a jejímu přečtení využívá příjemce agenta MUA. Tento agent kontaktuje server poskytující přístup k úložišti zpráv. Tento server je oddělen od MTA, který zprávu dodal, a je vytvořen

k zajišťování přístupu pro přebírání zpráv. Jakmile server žadatele úspěšně ověří, může zprávu tohoto uživatele odeslat jeho agentovi MUA. [9]



Obrázek 10 - Jednoduchý průchod zpráv internetem [10]

2.5.2 Protokoly elektronické pošty

Aby bylo možné docílit správné komunikace mezi poštovními agenty, je nutné použít protokoly, které nám danou komunikaci zprostředkují. K odesílání elektronických zpráv se využívá protokolu SMTP a k jejich příjmu slouží protokoly POP3 nebo IMAP. Tyto protokoly jsou definované organizací IETF a jsou specifikována pomocí RFC (Rewuest for commnets). [9,10]

2.5.2.1 Protokol SMTP

Protokol SMTP (Simple Mail Transfer Protokol) je jednoduchý internetový protokol, který je určený pro přenos elektronických zpráv z jednoho počítače na druhý. Aby byl tento přenos zajištěn, využívá SMTP síťový transportní protokol TCP, který se stará o spolehlivý přenos. Tento protokol používají pro odesílání zpráv jak klienti MUA, tak i samotné poštovní servery MTA mezi sebou. Protokol SMTP chápe přenášená data jako textová, členěná na jednotlivé řádky (pomocí znaků CR a LF) a tvořená pouze znaky z původní 128prvkové abecedy ASCII. SMTP předpokládá pouze přenos znaků kódovaných do sedmi bitů. Pokud se takové sedmibitové znaky přenášejí kanálem, který je uzpůsoben přenosu osmibitových znaků, pak standard SMTP definuje, že jeho sedmibitové znaky mají být vkládány do osmic bitů tak, aby byly zarovnány doprava a zleva doplněny nulovým bitem. [9,10]

2.5.2.2 Protokol POP

POP (Post Office Protocol) je internetový protokol, který slouží pro stahování elektronických zpráv ze vzdáleného úložiště zpráv na klienta. Tento protokol stejně, tak jako SMTP, využívá ke svému přenosu protokol TCP a naslouchá na portu 110. V současnosti je používána třetí verze POP3, která od starších verzí umožňuje značné množství nastavení, jako například možnost stáhnout pouze hlavičky zpráv. Celá komunikace probíhá v střídajících se výměnách mezi klientem serverem, skládá se z několika fází:

- **fáze spojení** – klient požádá o spojení, server pošle uvítání.
- **fáze autorizace** – server požaduje identifikaci klienta pro přístup do úložiště zpráv.
- **fáze transakční** – stažení a smazání elektronických zpráv.
- **fáze úprav** – provedení definovaných změn a ukončení spojení.

Po otevření TCP spojení začíná komunikaci server. Poté přichází spojení do fáze autorizace, ve které se musí klient serveru prokázat, že je oprávněn přistupovat k informacím. Po úspěšné autorizaci přichází na řadu transakční fáze, ve které probíhá informování o počtu zpráv, jejich stahování a mazání. Všechny změny v této fázi jsou pouze zaznamenávány, avšak nejsou prováděny. Po uzavření spojení, po němž nastává fáze úprav, ve které jsou všechny dříve prováděné změny reálně provedeny a zapsány na disk. Poté dochází k rozpojení spojení. Celá komunikace je bohužel nešifrována a tudíž ji může útočník lehce odposlechnout a zjistit, tak přihlašovací jméno a heslo. Abychom tuto skutečnost zamezili, je nutné použít nějaký druh šifrování, například SSL nebo modernějšího TLS. [9,10]

2.5.2.3 Protokol IMAP

Protokol IMAP (Internet Message Access Protocol) v současné době IMAP verze 4 je podobně jako POP3 určen ke stahování elektronických zpráv z úložiště zpráv na klienta. Na rozdíl od POP3 je optimalizován pro práci s elektronickými zprávami v režimu dlouhého připojení. Je mnohem složitější a nabízí mnohem větší komfort pro práci se zprávami. Na rozdíl od POP3, v němž se zprávy stahují okamžitě ze serveru na klienta, jsou zprávy permanentně uloženy na serveru. Díky této možnosti může se zprávami pracovat více poštovních klientů. Klient může zprávy přesouvat mezi schránkami, editovat je, ukládat či načítat. Je také možné nastavit stahování pouze záhlaví zpráv nebo celé zprávy. Každá zpráva obsahuje značku, která ukazuje, zda byla zpráva již přečtena, či smazána. Tyto značky se automaticky synchronizují mezi použitými poštovními klienty. I u tohoto protokolu je nutné komunikaci šifrovat pomocí SSL/TLS, abychom zabránili případnému odposlechu spojení. [9,10]

2.5.3 SMTP server Postfix

Pro operační systém Linux existuje hodně SMTP serverů (MTA agentů), jako jsou například Exim, Sendmail, Gmail, apod. Pro potřeby diplomové práce jsem zvolil Postfix, jelikož se jedná o

nejpoužívanější a velmi oblíbené řešení SMTP serveru, které je poskytováno pod licencí IBM Public License a jeho autorem je Wietse Venema. Velkou výhodou Postfixu je jednoduchá instalace, konfigurace a také modulární architektura, díky které lze Postfix snadno modifikovat a rozšiřovat o další funkce. Návrh vývoje Postfixu je zaměřen na určité cíle, mezi které patří [9,10]:

- **Spolehlivost** – při práci ve vysokém zatížení, kdy se většina softwarových systémů chová nepředvídatelně, když jim dojde paměť či diskový prostor. Postfix detekuje takové chování a nabídne systému možnost „vzpamatovat se“. Postfix se snaží všemožnými způsoby fungovat stabilně a spolehlivě.
- **Zabezpečení** – k ochraně proti útočníkům je zavedeno několik obranných vrstev. Každý proces, který může běžet izolovaně, běží s nejnižší sadou oprávnění. Procesy s vyšším oprávněním nikdy nedůvěřují neoprávněným procesům. Moduly, které nepotřebujeme, lze snadno deaktivovat, čímž se nám zvýší zabezpečení a zjednoduší instalace.
- **Výkon** – postfix byl vytvořen s ohledem na dosažení vysokého výkonu, ale s předpokladem, že jeho rychlost neomezí jiné systémy. Speciálními technikami omezuje, jak počet nových procesů, které je nutno vytvořit, tak i počet přístupů k systému souborů, jež jsou zapotřebí v rámci zpracování zpráv.
- **Flexibilita** – systém Postfix je složen z několika různých programů a podsystémů. Tímto přístupem je možno dosáhnout vysoké flexibility. Všechny části lze upravovat pomocí jednoduchých konfiguračních souborů.
- **Snadné používání** – z hlediska nastavení a správy je Postfix jedním z nejjednodušších balíčků určených pro zpracování elektronických zpráv. Pracuje s jednoduchými konfiguračními soubory a vyhledávacími tabulkami zajišťujícími překlad adres a předávání zpráv.

2.5.4 POP3/IMAP server Dovecot

Stejně jako existuje celá řada SMTP serverů, tak existuje i celá řada POP3/IMAP serverů. Některé podporují oba poštovní protokoly a některé jen jeden z nich. Mezi nejoblíbenější patří bezpochyby Courier, Cyrus a Dovecot. Dovecot je poměrně mladý a často využívaný pro podporu POP3 a IMAP, při jeho vývoji kladli vývojáři velký důraz na bezpečnost a další vlastnosti, které jsou podobné jako u Postfixu, právě proto se tyto dva programy k sobě skvěle hodí. To byl také hlavní důvod, proč jsem pro podporu protokolů POP3 a IMAP zvolil právě Dovecot. Architektura by se dala shrnout takto [9,10]:

- **Podpora standardů** – hodně dostupných IMAP serverů (u POP3 serverů je situace jiná vzhledem k primitivnosti protokolu oproti protokolu IMAP) trpí většími či menšími

odchylkami od specifikace protokolu IMAP4rev1. Dovecot úspěšně prošel všemi ze 403 skriptovacích testů zaměřených na základní IMAP a dalšími 100 testy na rozšíření protokolu. Dovecot navíc podporuje různé workaroundy, kterými se vyrovná s chybami v implementaci protokolu na straně klientů.

- **Stabilita a bezpečnost** – stejně jako u Postfixu je řešena bezpečnost i zde, a to díky modularitě, minimalizací a separací práv atd. Moduly jsou nezávislé a problémy jednoho nemají dopady na fungování ostatních.
- **Flexibilita** – Dovecot si lze jednoduše přizpůsobit, a to jak z hlediska úložiště zpráv, tak i datových zdrojů, autentizačních metod, atd. Snadná je i migrace z většiny jiných POP3/IMAP řešení.

3 Realizace služeb

3.1 Virtualizace v ROS

ROS nativně nepodporuje služby, jako jsou poštovní či webový server, ale umožňuje nám tyto služby realizovat jiným způsobem, a to díky hardwarové virtualizaci KVM, která je v tomto systému implementována. V následujících krocích předvedu, jak v ROS zavést virtuální systém, tj. hostovaný operační systém, který je nainstalován a spuštěn ve virtuálním počítači, který nám umožní běh již zmíněných služeb. Dříve, než se pustíme do vytvoření hostovaného systému, je nutné nějakým způsobem do ROS nahrát obraz disku operačního systému, který chceme virtualizovat, což můžeme provést buď pomocí ftp klienta, nebo - pokud používáme winbox - jednoduchým uchopením obrazu operačního systému a jeho přenesením na pozadí winboxu. Veškeré konfigurace, které budu provádět, je možné provést jak pomocí CLI, tak pomocí winboxu.

3.1.1 Vytvoření hostovaného systému

Aby bylo možné provozovat hostovaný systém v ROS, je nutné nejprve vytvořit obraz virtuálního disku, na který se bude náš operační systém instalovat. Virtuální disk vytvoříme pomocí příkazu [2]:

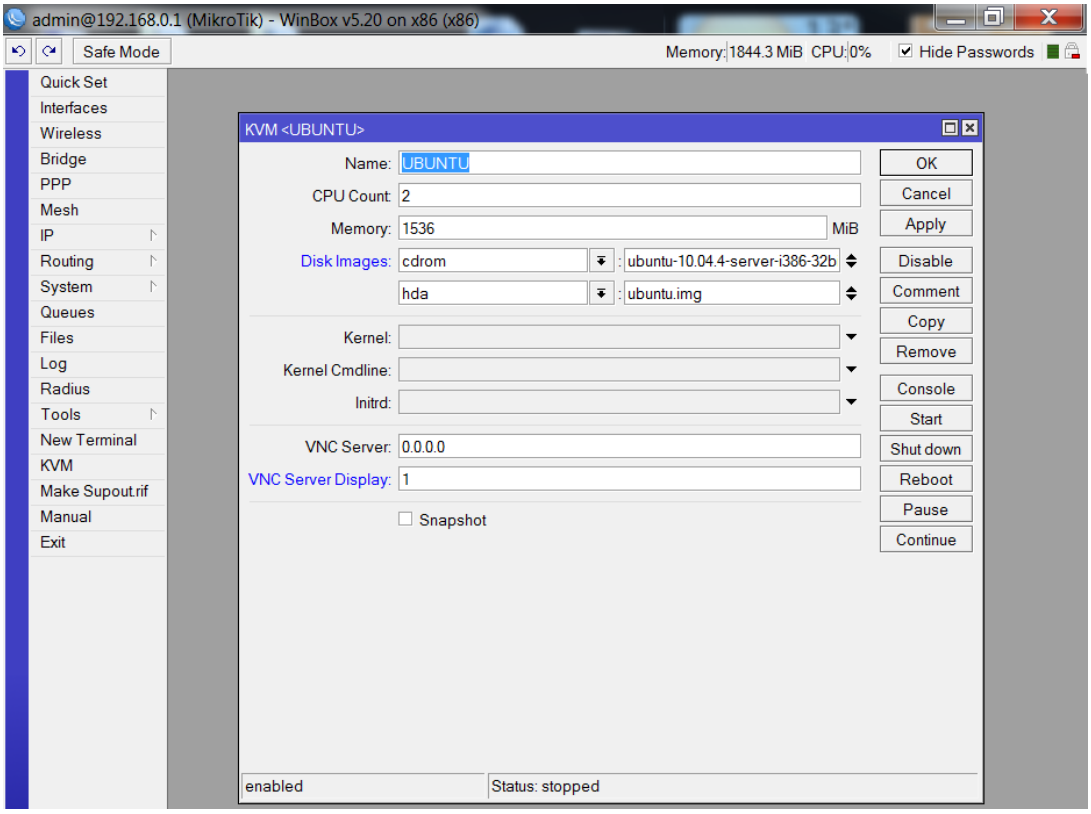
- `kvm make-routeros-image file-name=ubuntu.img file-size=10000`

Po úspěšném vytvoření *10 GB* obrazu virtuálního disku s názvem *ubuntu.img* je dále nutné specifikovat parametry pro virtualizované prostředí, ve kterém poběží náš hostovaný systém.

- `kvm add name=UBUNTU memory=1536 cpu-count=2 disable=no disk-image=cdrom:ubuntu-10.04.4-server-i386-32bit.iso,hda:ubuntu.img initrd="" kernel="" kernel-cmdline="" vnc-server-address=0.0.0.0 vnc-server-dispaly=1 comment=UBUNTU-VIRTUALNI-STROJ`

Nyní jsme úspěšně vytvořili virtuální stroj „*UBUNTU*“, kterému jsme přiřadili *1536 MB* operační paměti a povolili mu přístup k využívání obou jader procesoru CPU. Parametrem *disable=no* zakážeme, aby se virtuální stroj po jeho vytvoření spustil. Parametr *disk-image:cdrom:ubuntu-10.04.4-server-i386-32bit.iso* slouží pro zavedení instalačního image operačního systému, který jsme již do ROS přenesli (po nainstalování operačního systému se musí virtuální disk CD-ROM odebrat), dále parametrem *hda:ubuntu.img* definujeme použití virtuálního obrazu, jako Primary Master IDE. Pro

vzdálený přistup do hostovaného systému je využit VNC server s nastaveným portem *1*. Další parametry v našem případě již nejsou povinné. [2]



Obrázek 11 - nastavení KVM

Dalším krokem je nutné vytvořit síťové rozhraní, tj. síťový most, který nám zajistí komunikaci mezi fyzickým a virtuálním rozhraním. Vytvoření síťového mostu *„kvm_ubuntu“* provedeme pomocí příkazu [2]:

- interface bridge add name=kvm_ubuntu

V posledním kroku vytvoříme virtuální rozhraní náležící virtuálnímu stroji *„UBUNTU“* a přidáme jej do síťového mostu *„kvm_ubuntu“*. Do tohoto síťového mostu následně přidáme fyzické rozhraní *„ether1“* náležící ROS. [2]

- `kvm interface add virtual-machine=UBUNTU type=dynamic dynamic-bridge=kvm_ubuntu`

- `interface bridge port add bridge=kvm_ubuntu interface=Eth1`

V tuto chvíli máme zajištěnou komunikaci mezi virtuálním strojem *„UBUNTU“* a fyzickým rozhraním *„ether1“* náležícímu ROS. V této fázi můžeme přistoupit ke spuštění virtuálního stroje pomocí příkazu: [2]

- kvm start UBUNTU

3.2 Instalace GNU/Linux Ubuntu server

Samotná instalace Ubuntu serveru je velmi jednoduchá a skládá se z několika kroků.

V prvních krocích jsme dotázáni na obecné údaje, jako jsou jazyková lokalizace a rozložení klávesnice, poté následuje automatická konfigurace sítě pomocí DHCP. Pokud se operace nezdaří, nabídne nám instalátor možnost nakonfigurovat síť ručně, nebo síť prozatím nenastavovat. Zvolíme možnost „prozatím síť nenastavovat“. V dalším kroku jsme dotázáni na zadání jména počítače.

Pak následuje rozdělení pevného disku. Je zde několik možností výběru. Je možné manuálně nastavit velikost diskových oblastí, včetně výběru souborového systému, nebo použití asistované rozdělení, přičemž na výběr máme ze tří možností, a to: použít celý disk, použít celý disk a nastavit LVM, použít celý disk a nastavit šifrované LVM. LVM představuje vrstvu nad fyzickými zařízeními, která umožňuje za běhu měnit velikost logických oddílů. Pro tuto práci nemá toto nastavení žádný vliv, a proto je na místě volba „asistované – použít celý disk“.

Po provedení diskových operacích se nám systém nainstaluje a následně jsme dotázáni na jméno a heslo pro nového uživatele.

V předposledním kroku jsme dotázání na aktualizace systému. Zvolíme možnost „žádné automatické aktualizace“. Poté nás systém vyzve, zda chceme instalovat nadefinované programy, jako jsou například DNS server, Print server, atd. Ponecháme systém čistý, potřebné programy nainstalujeme později.

V posledním kroku potvrdíme nainstalování zavaděč GRUB do hlavního zaváděcího záznamu MBR disku. V této fázi je instalace Ubuntu server kompletní.

Obrázek 12 - spuštění hostovaného operačního systému

3.2.1 Přihlášení se do systému

Po každém spuštění operačního systému Ubuntu server jsme dotázání na vyplnění jména a hesla pro přístup do operačního systému. Tyto údaje jsme vyplnili při instalaci, tudíž máme plný přístup. Jelikož některé operace, které se budou provádět, přesahují práva běžného uživatele, je nutné využít práva superuživatele ROOT. Abychom mohli využívat těchto práv, je nutné se jako superuživatel přihlásit, a to provedeme následovně pomocí příkazu *sudo su* a zadáním hesla uživatele, který byl vytvořen při instalaci systému. Následně pomocí příkazu *passwd root* vytvoříme nové heslo pro uživatele ROOT. Veškeré nastavení a konfigurace v této práci jsem vytvářel pouze pod účtem ROOT.

3.2.2 Nastavení sítě

V průběhu instalace nebyla síť nastavena. Její nastavení provedeme jednoduchou editací souboru */etc/network/interface* a to pomocí příkazu:

- `nano /etc/network/interface`

Do konfiguračního souboru přidáme následující:

```
auto eth0
iface eth0 inet static
      address 192.168.0.2
      netmask 255.255.255.0
      gateway 192.168.0.1
```

Servery DNS se konfigurují v souboru */etc/resolv.conf*

- `nano /etc/resolv.conf`

Do konfiguračního souboru vložíme potřebné DNS servery:

```
nameserver 192.168.0.2
nameserver 192.168.0.1
```

Následně zapíšeme do souboru */etc/hosts* IP adresy, které patří našemu serveru. Zde můžeme zadat jména všech strojů, které nechceme zjišťovat u DNS, a definujeme jejich IP adresy přímo. Můžeme nadefinovat pro jeden stroj i několik aliasů.

- `nano /etc/hosts`

Konfigurační soubor upravíme následovně:

```
127.0.0.1       localhost.localdomain      localhost
192.168.0.2     server.mojedomena.cz       server
```

Poslední úpravou, která se provede, je modifikace souboru */etc/hostname*. Zde můžeme nastavit plně specifikované doménové FQDN jméno.

- `nano /etc/hostname`

Do konfiguračního souboru přidáme: `server.mojedomena.cz`

Nyní k promítnutí všech změn je nutné služby restartovat, a to pomocí následujících příkazů:

- `/etc/init.d/networking restart`
- `/etc/init.d/hostname.sh restart`

Následně můžeme ověřit, zda se nám změna hostname podařila. Je nutné, aby doménové jméno bylo stejné jako FQDN jméno. K tomuto ověření slouží následující příkazy:

- `hostname`
- `hostname –f`

3.2.3 Aktualizace systému

Abychom mohli instalovat potřebné balíčky, je nutné aktualizovat lokální databázi balíčků z repozitáře. To se provede příkazem *update* a až příkazem *upgrade* provedeme aktualizaci jednotlivých balíčků.

- `apt-get update`
- `apt-get upgrade`

3.2.4 Vzdálený přístup

Abychom nemuseli neustále přistupovat ke konfigurování Ubuntu server pomocí utility VNC, nabízí se nám použití daleko lepšího řešení v podobně OpenSSH. Toto řešení nám zajistí vzdálený přístup k serveru, které je navíc šifrováno. Instalaci provedeme příkazem:

- `apt-get install openssh-server ssh`

3.3 Realizace webového serveru

Samotná instalace webového serveru na Ubuntu server je velmi rychlá a snadná. Pomocí jednoho příkazu se nám nainstalují všechny potřebné balíčky pro jeho plnou realizaci. Nainstaluje se nám plnohodnotný LAMP server. Při instalaci balíčků jsme dotázáni pouze na vytvoření hesla pro superuživatele ROOT pro přístup do MySQL databáze. Instalace LAMP se provede pomocí příkazu:

- `apt-get install lamp-server^`

Pro představu se nám nainstalují následující balíčky:

- `apache2 apache2-mpm-prefork apache2-utils apache2.2-bin apache2.2-common libapache2-mod-php5 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libdbd-mysql-perl libdbi-perl libhtml-template-perl libmysqlclient16 libnet-daemon-perl libplrpc-perl mysql-client-5.1 mysql-client-core-5.1 mysql-common mysql-server mysql-server-5.1 mysql-server-core-5.1 php5-common php5-mysql ssl-cert`

V této fázi máme nainstalovaný LAMP. Pro ještě jednodušší práci s databází můžeme dále nainstalovat hojně využívanou aplikaci phpMyAdmin. Při její instalaci jsme dotázáni na výběr

webového serveru, v našem případě vybereme *apache2,* následně potvrdíme nastavení databáze pomocí *dbconfig-common*. Tato funkce nám usnadní konfiguraci. Následuje zadání hesla pro přístup do MySQL databáze a v závěru vytvoříme heslo pro přístup do aplikace phpMyAdmin. Instalace se provede zadáním příkazu:

- `apt-get install phpmyadmin`

3.3.1 Ověření funkčnosti

V této fázi máme nainstalováno vše potřebné a je nutné ověřit, zda nám vše funguje, jak má. V prvním kroku ověříme správnou funkčnost Apache, následně PHP a závěrem phpMyAdmin.

- **Test Apache** - do našeho internetového prohlížeče zadáme adresu http://192.168.0.2/ Výstupem testu by měla být věta ve smyslu: „It works!“.
- **Test PHP** – vytvoříme soubor do adresáře */var/www* s názvem *test.php*. Obsahem souboru bude následující řádek kódu: *<?php phpinfo(); ?>* Poté provedeme restartování služby */etc/init.d/apache2 restart*. Po zadání adresy http://192.168.0.2/test.php bude výstupem stránka s obsahem verze PHP a dalších věcí.
- **Test phpMyAdmin** – po zadání adresy http://192.168.0.2/phpmyadmin/ bychom měli být schopni se do této aplikace přihlásit.

3.4 Realizace poštovního serveru

Instalace poštovního serveru je taktéž velmi snadná. Vše, co budeme k realizaci poštovního serveru potřebovat, opět nainstalujeme pomocí jednoho příkazu.

- `apt-get install mail-server^`

Při instalaci balíčků jsme dotázáni na několik otázek. První otázka zní, jaký typ varianty poštovního serveru nejlépe odpovídá našim požadavkům. Vybereme „Internetový počítač“ - to znamená, že elektronická pošta bude odesílána a přijímána přímo protokolem SMTP. Dalším krokem je zadání doménového jména serveru. Již v základní konfiguraci jsme nastavili doménové jméno na *server.mojedomena.cz,* a mělo by tedy být uvedeno i zde.

Díky dnešním trendům v realizaci poštovních serverů jsem se rozhodl využít MySQL databázi pro vedení uživatelských účtů, i když řešení se schránkami běžných uživatelů v operačním systému je velmi jednoduché a v řadě případů dobře použitelné. Použití virtuálních uživatelů má celou řadu výhod jako jsou:

- možnost pracovat s poštou většího počtu domén,
- možnost mít bezpečně uložená hesla,
- možnost spravovat zcela samostatně databázi uživatelů a hesel,
- o přístupová práva se stará poštovní server.

Pro jednoduchou správu virtuálních poštovních schránek jsem zvolil aplikaci postavenou na licenci GNU s názvem Postfixadmin. Tato aplikace bohužel není obsažena v repozitářích, a proto je nutné ji stáhnout z webové stránky vývojářů. Pro správný chod této aplikace je nutné stáhnout také několik balíčků, na kterých je Postfixadmin závislý.

- `apt-get install dbconfig-common libc-client2007e mlock php5-imap wwwconfig-common postfix-mysql sasl2-bin`

Stažení aplikace Postfixadmin provedeme následovně:

- `wget http://sourceforge.net/projects/postfixadmin/files/postfixadmin/postfixadmin-2.3.5/postfixadmin_2.3.5-1_all.deb/ -O postfixadmin_2.3.5-1_all.deb`

Poté pomocí následujícího příkazu provedeme nainstalování aplikace:

- `dpkg -i postfixadmin_2.3.5-1_all.deb`

Při instalaci jsme dotázáni na výběr webového serveru, který chceme použít - zvolíme *apache2*. Potvrdíme nastavení databáze pomocí *dbconfig-common* a zvolíme použití databáze *MySQL*. Vzápětí jsme dotázáni na heslo pro vstup do MySQL a heslo pro balík Postfixadmin.
V této chvíli máme nainstalováno vše, co budeme k realizaci poštovního serveru potřebovat. Nyní přejdeme k další náročnější části, kterou představují jednotlivé konfigurace.

3.4.1 Konfigurace aplikace Postfixadmin

Konfigurace aplikace Postfixadmin se provádí v souboru */etc/postfixadmin/config.inc.php*. V tomto souboru nastavíme, aby Postfixadmin přistupoval do databáze MySQL.

- `nano /etc/postfixadmin/config.inc.php`

Konfigurační soubor upravíme následovně:

```
$CONF['database_type'] = 'mysqli';
$CONF['database_host'] = 'localhost';
$CONF['database_user'] = 'postfixadmin';
$CONF['database_password'] = 'heslo_do_database';
$CONF['database_name'] = 'postfixadmin';
```

Nyní spustíme webový prohlížeč s adresou *http://192.168.0.2/postfixadmin/setup.php* a budeme vyzváni ke změně hesla. Poté se nám vygeneruje hash, který vložíme do téhož konfiguračního souboru. V tomto kroku se nám také zároveň v databázi vytvoří tabulky, které slouží pro ukládání parametrů o uživatelích, doménách apod.

```
$CONF['setup_password'] = 'hash';
```

V této chvíli máme nastavený Postfixadmin a můžeme vytvořit Administrátora, který bude řídit správu poštovního serveru.

Obrázek 13 - Administrace v aplikaci Postfixadmin

3.4.2 Konfigurace Postfixu

Veškeré nejdůležitější konfigurace Postfixu se provádějí ve dvou konfiguračních souborech, a sice v *main.cf* a *master.cf*. Soubor *master.cf* představuje hlavní modul Postfixu, který je složen z výkonných modulů realizujících konkrétní úlohy. Soubor *main.cf* je jádrem celého Postfixu. Téměř všechny konfigurační změny se provádějí právě zde. Tento soubor můžeme snadno editovat pomocí příkazu *postconf –e,* nebo opět pomocí textového editoru.

Základní konfiguraci souboru *main.cf* provedeme příkazem:

- `nano /etc/postfix/main.cf`

V tomto souboru je nutné říci Postfixu, kde a jak má hledat uživatele, kteří budou poštu přijímat, do kterých domén má doručovat a kde se nachází úložiště konkrétního uživatele. Proto je nutné do tohoto souboru přidat příslušné mapy a jim přiřadit mapovací soubory. Tyto mapovací soubory obsahují informace, díky kterým je možné získat požadované hodnoty z databáze.

Do tohoto souboru přidáme následující řádky:

```
virtual_mailbox_domains = mysql:/etc/postfix/mailbox_domains.cf
virtual_mailbox_maps = mysql:/etc/postfix/mailbox_maps.cf
virtual_alias_maps = mysql:/etc/postfix/alias_maps.cf
relay_domains = mysql:/etc/postfix/relay_domains.cf
```

Dále je nutné vytvořit mapovací soubory pro každou zde vypsanou mapu. Prvním vytvořeným souborem bude *mailbox_domain.cf.*

- `nano /etc/postfix/mailbox_domains.cf`

Obsahem tohoto souboru budou následující řádky:

```
hosts = 127.0.0.1
user = postfixadmin
password = HESLO
dbname = postfixadmin
query = SELECT domain FROM domain WHERE domain='%s' and
backupmx = 0 and active = 1
```

Vytvoření souboru *mailbox_maps.cf*

- `nano /etc/postfix/mailbox_maps.cf`

Obsah souboru:

```
hosts = 127.0.0.1
user = postfixadmin
password = HESLO
dbname = postfixadmin
query = SELECT maildir FROM mailbox WHERE username='%s' AND
active = 1
```

Vytvoření souboru *alias_maps.cf*

- `nano /etc/postfix/alias_maps.cf`

Obsah souboru:

```
hosts = 127.0.0.1
user = postfixadmin
password = HESLO
dbname = postfixadmin
query = SELECT goto FROM alias WHERE address='%s' AND active =1
```

Vytvoření souboru *relay_domains.cf*

- `nano /etc/postfix/relay_domains.cf`

Obsah souboru:

```
hosts = 127.0.0.1
user = postfixadmin
password = HESLO
```

```
dbname = postfixadmin
query = SELECT domain FROM domain WHERE domain='%s' and
backupmx = 1
```

V této fázi je nutné zvolit metodu, jakou se budou poštovní zprávy ukládat. Jelikož uživatelé poštovního serveru jsou virtuální, čili neexistují v operačním systému, bude veškerá pošta zpracována pod jedním uživatelským účtem, který je nutno založit. Nejprve vytvoříme skupinu *vmail,* které přiřadíme hodnotu GID *5000,* a poté vytvoříme uživatele *vmail* s domovským adresářem */home/vmail* a hodnotami UID *5000* a GID *5000.* Do tohoto adresáře se budou nyní vytvářet poštovní schránky.

- `groupadd -g 5000 vmail`
- `useradd -d /home/vmail -m -u 5000 -g 5000 vmail`

Nyní můžeme do konfiguračního souboru *main.cf* přidat hodnoty UID a GID, se kterými nakládá doručovací agent. Dále sdělíme Postfixu místo, kde nalezne poštovní schránky virtuálních uživatelů.

- `postconf -e 'virtual_uid_maps = static:5000'`
- `postconf -e 'virtual_gid_maps = static:5000'`
- `postconf -e 'virtual_mailbox_base = /home/vmail'`

V tuto chvíli máme nakonfigurovaný základ Postfixu. Poslední věcí, kterou je nutno ještě provést, je jej zabezpečit, abychom předešli neoprávněným uživatelům, kteří by mohli skrze náš poštovní server odesílat elektronickou poštu. Mechanismů pro zabezpečení existuje vícero. Já jsem zvolil autentizační metodu zvanou SASL (Simple Authentication and Security Layer), která řeší ověřování uživatelů pomocí jména a hesla.

Tuto ověřovací technologii aktivujeme přidáním následujícího řádku:

- `postconf -e 'smtpd_sasl_auth_enable = yes'`

Pomocí příkazu *postconf –a* zjistíme, jaký typ poskytovatele SASL Postfix podporuje. V mém případě podporuje Cyrus a Dovecot. Jelikož budu používat Dovecot pro podporu POP3/IMAP, volím jako autentizačního agenta právě jeho. Aby Postfix využíval autentizační služby je nutné je aktivovat:

- `postconf –e 'smtpd_sasl_type = dovecot'`

Dále je nutné specifikovat cestu k socketu pro komunikaci.

- `postconf –e 'smtpd_sasl_path = private/auth'`

Poslední věcí, kterou je potřeba udělat, je definovat pravidla pro restrikce přístupu tak, aby bylo doručování do místních schránek povoleno odkudkoliv a odesílání do internetu pouze autentizovaným uživatelům. To provedeme pomocí následujících řádků:

- `postconf –e 'smtpd_recipient_restrictions =`
  `permit_mynetworks,`
  `permit_sasl_authenticated,`
  `reject_non_fqdn_hostname,`
  `reject_non_fqdn_sender,`
  `reject_non_fqdn_recipient,`
  `reject_unauth_destination,`
  `reject_unauth_pipelining,`
  `reject_invalid_hostname'`

3.4.3 Konfigurace Dovecotu

Samotná konfigurace Dovecotu je oproti Postfixu o poznání snazší. Konfigurace se provádějí v souboru */etc/dovecot/dovecont.conf*. Veškeré změny v tomto konfiguračním souboru budu opět provádět pomocí textového editoru *nano*. Soubor opět používá formát *parametr = hodnota* s tím, že některé části konfigurace uzavírá mezi složené závorky. Dobrou vlastností Dovecotu je, že může fungovat jako lokální doručovací agent LDA (Local Delivery Agent), který může přebírat zprávy od MTA a zajišťovat finální doručení do schránek. Použití tohoto řešení má mnoho výhod oproti použití Postfixu. Je dobré, když elektronickou poštu umístí do schránky stejný program, který ji pak bude pro uživatele vyzvedávat. Další výhodou je modularita - je možné si zvolit různé zásuvné moduly pro filtrování pošty, jako je například modul sieve.

Aby bylo možné Dovecot používat jako lokálního doručovacího agenta, je nutné přidat tuto službu do konfiguračního souboru */etc/postfix/mater.cf*, kde bude naslouchat na unixovém socketu a bude spouštět program *deliver* pod uživatelem *vmail*, který je určený pro přístup do úložiště zpráv. Toto nastavení provedeme následovně:

- `nano /etc/postfix/master.cf`

Vložíme následující obsah:

```
dovecot unix    -    n    n    -    -    pipe
flags=DRhu user=vmail:vmail argv=/usr/lib/dovecot/deliver -d
${recipient}
```

Dále je nutno říci Postfixu, která služba se použije pro doručování virtuálním uživatelům a že se bude doručovat pouze jedinému příjemci v rámci jedné instance.

- `postconf –e 'virtual_transport = dovecot'`
- `postconf –e 'dovecot_destination_recipient_limit = 1'`

Další změny budeme provádět již v samotném konfiguračním souboru *dovecont.conf.* Jelikož budeme používat Dovecot LDA, je nutné vykonat několik konfigurací pro správný běh. Aby program *deliver* věděl, kam poštu posílat, musíme upravit konfigurační soubor následovně:

```
userdb static {
args=uid=5000 gid=5000 home=/home/vmail/%d/%n
allow_all_users=yes
}
```

Nyní je potřeba nakonfigurovat rozhraní pro doručovacího agenta *master* a zároveň nakonfigurujeme rozhraní *client*, aby Postfix využíval pro autentizaci uživatelů autentizačního agenta Dovecot. To provedeme následovně:

```
socket listen {
master {
path = /var/run/dovecot/auth-master
mode = 0600
user = vmail
 }

client {
path = /var/spool/postfix/private/auth
mode = 0660
user = postfix
group = postfix
 }
}
```

Konečně můžeme přistoupit ke konfiguraci samotného doručovacího agenta LDA, který nabízí více možností oproti vestavěnému Postfixu. Jeho konfigurace je snadná. Pouze určíme cestu k ověřovacímu serveru a vytvoříme poštovní adresu určenou pro posílání systémových zpráv o odmítnutých zprávách.

```
protocol lda {
auth_socket_path = /var/run/dovecot/auth-master
postmaster_address = postmaster@mojedomena.cz
}
```

Nyní povolíme protokoly, které chceme využívat pro komunikaci s poštou.

```
protocols = imap pop3
```

Jelikož realizovaný poštovní server neřeší šifrovanou komunikaci SSL/TLS, je potřeba zakázat vynucení šifrované komunikace.

```
disable_plaintext_auth = no
```

Dále je nutné specifikovat, kde se poštovní schránky virtuálních uživatelů nachází:

```
mail_location=maildir:/home/vmail/%d/%n/Maildir:INDEX=/home/vma
il/%d/%n/Maildir/indexes
```

Také musíme Dovecotu definovat přístup do databáze MySQL pro autorizaci uživatelů (*passdb* specifikuje databázi hesel a *userdb* databázi, kde se o uživateli získá zbytek informací).

```
auth default {
userdb sql {
args = /etc/dovecot/dovecot-mysql.conf
     }
passdb sql {
args = /etc/dovecot/dovecot-mysql.conf
     }
}
```

Nyní musíme soubor *dovecot-mysql.conf* vytvořit:

- `nano /etc/dovecot/dovecot-mysql.conf`

Obsahem tohoto souboru budou nakonfigurované databázové dotazy pro ověření hesla a zjištění údajů o uživateli. Soubor vypadá následovně:

```
driver = mysql
connect = host=127.0.0.1 dbname=postfixadmin user=postfixadmin
password=HESLO
default_pass_scheme = MD5-CRYPT
user_query = SELECT '/home/vmail/%d/%n' as home, 5000 AS uid,
5000 AS gid FROM mailbox WHERE username = '%u'
password_query = SELECT password FROM mailbox WHERE username =
'%u'
```

V tuto chvíli máme již konfiguraci celého poštovního serveru hotovou. Nezbývá nám již nic jiného než služby restartovat a ověřit, zda funguje všechno, jak má.

- `/etc/init.d/dovecot restart`
- `/etc/init.d/postfix restart`

3.5 Realizace stavového firewallu

V této kapitole bude realizováno základní zabezpečení hraničního směrovače pomocí stavového firewallu operačního systému ROS.

První akcí, kterou provedeme, je zamaskování naší lokální sítě 192.168.0.0/24 za jednu veřejnou IP adresu, kterou máme přidělenou od poskytovatele internetového připojení.

Aplikace pravidla v tabulce NAT

- `ip firewall nat add chain=srcnat action=masquerade out-interface=Eth0 comment=Masquerade`

Dále provedeme přesměrování protokolů SMTP, POP3/IMAP a HTTP na náš hostovaný operační systém Ubuntu.

Aplikace pravidel pro tabulku NAT a řetězec DSTNAT

- `ip firewall nat add chain=dstnat dst-address=10.0.0.2 protocol=tcp dst-port=25 action=dst-nat to-addresses=192.168.0.2 to-ports=25 comment=SMTP`
- `ip firewall nat add chain=dstnat dst-address=10.0.0.2 protocol=tcp dst-port=110 action=dst-nat to-addresses=192.168.0.2 to-ports=110 comment=POP3`
- `ip firewall nat add chain=dstnat dst-address=10.0.0.2 protocol=tcp dst-port=143 action=dst-nat to-addresses=192.168.0.2 to-ports=143 comment=IMAP`
- `ip firewall nat add chain=dstnat dst-address=10.0.0.2 protocol=tcp dst-port=80, 443 action=dst-nat to-addresses=192.168.0.2 to-ports=80 comment=HTTP`

Aplikace pravidel pro tabulku FILTER a řetězec INPUT

Pakety, které nejsou součástí známého spojení, budou zahozeny.

- `ip firewall filter add chain=input connection-state=invalid action=drop comment=Zakazat_neplatne_spojeni`

Pakety od již navázaných spojení budou povoleny.

- `ip firewall filter add chain=input connection-state=established action=accept comment=Povolit_navazana_spojeni`

Povolení protokolu ICMP

- `ip firewall filter add chain=input protocol=icmp action=accept comment=Povoleni_ICMP`

Povolení protokolu SSH

- `ip firewall filter add chain=input protocol=tcp dst-port=22 action=accept comment=Povoleni_SSH`

Povolení přístupu z lokální sítě

- `ip firewall filter add chain=input src-address=192.168.0.0/24 action=accept in-interface=!Eth0 comment=Povoleni_Lan`

Zakázání veškerého spojení

- `ip firewall filter add chain=input action=drop comment=Zakaz_veskereho_spojeni`

Aplikace pravidel pro tabulku INPUT a řetězec FORWARD

Povolení provozu pro již navázaná spojení

- `ip firewall filter add chain=forward connection-state=established action=accept comment=Povoleni_jiz_navazanych_spojeni`

Povolení provozu pro navazující spojení

- `ip firewall filter add chain=forward connection-state=related action=accept comment=Povoleni_navazujicich_spojeni`

Pakety, které nejsou známého spojeni, budou zahozeny

- `ip firewall filter add chain=forward connection-state=invalid action=drop comment=Zakazat_neplatne_spojeni`

4 Srovnávací testy služeb

V této části diplomové práce jsem objektivně porovnal virtualizované služby se službami nativními. V první části je popsána metodika testování, dále použité nástroje, díky nimž bylo testování realizováno, a v závěru kapitoly se nacházejí samotné výsledky s vyhodnocením.

4.1 Metodika testování

Pro testování provozovaných služeb neexistují žádné konkrétní normy, dle kterých by bylo možné testy provádět. Srovnání služeb jsem pojal jako výkonnostní testování, to znamená, jak velký provoz je schopna služba obstarat, zda při testování nedochází k pádům služby apod. Veškeré testování probíhalo po síťové komunikaci o rychlosti 1Gb/s, jelikož testovací nástroje, které byly zvoleny, by ovlivňovaly výkon systému, a tak by nebyl test zcela průkazný. Pro výkonové testování webového serveru jsem zvolil nástroj Apache benchmark, pro testování poštovního serveru byl zvolen SMTP generátor SMTP-source spolu s linuxovými nástroji time a top. Konfigurace poštovního a webového serveru nebyly pro testování nijak modifikovány a byly ponechány v defaultním nastavení.

4.1.1 Měření Webového serveru

Apache benchmark

Jedná se o testovací nástroj, který se nazývá „ab“ a je součástí webového serveru Apache. Díky tomuto nástroji je možné simulovat chování serveru při zátěži. Program si klade za cíl zjistit, kolik požadavků za jednotku času dokáže daný server obstarat. Pro testování byla zvolena statická webová stránka o velikosti 177 bytů. Celkový počet odeslaných požadavků byl neměnný a byl nastaven na hodnotu 10000. Byla měněna pouze konkurence vyslaných požadavků, tj. počet paralelně vyslaných požadavků, a to od 10 současných spojení až po 1000 současných spojení.

Testování webového serveru probíhalo následujícím způsobem:
ab – n 10000 –c <10/100/1000> *http://192.168.0.2/index.html*

- kde n 10000 udává celkový počet odeslaných požadavků
- kde c <10/100/1000> udává počet paralelně vyslaných požadavků

Zhodnocení naměřených výsledků

V tabulkách číslo 4 a číslo 5 jsou uvedeny výsledné hodnoty měření. Z naměřených hodnot je zřejmé, že s narůstajícím počtem souběžných požadavků dosahuje doba průběhu testu vyšších hodnot a přenosová rychlost neúměrně klesá. Dále si můžeme povšimnout, že v případě nasazení webového serveru jako nativní služby je dosaženo o poznání lepších výsledků než u služby, která je v ROS virtualizována. Při nastavení 10000 dotazů a 1000 paralelních přístupů dosahovala doba průběhu testu u služby nativní lepších hodnot než u služby virtualizované při použití 10 souběžných požadavků, totéž platí i pro přenosovou rychlost.

Tabulka 4 – Naměřené hodnoty webového serveru – nativní služba

Počet souběžných požadavků	Doba průběhu testu [s]	Přenosová rychlost [kbytes/s]
10	0,887	4987,24
100	0,919	4815,77
1000	1,499	2950,52

Tabulka 5 - Naměřené hodnoty webového serveru – virtualizovaná služba

Počet souběžných požadavků	Doba průběhu testu [s]	Přenosová rychlost [kbytes/s]
10	1,641	2696,20
100	1,913	2311,99
1000	3,398	1301,96

4.1.2 Měření Poštovního serveru

Smtp-source

Smtp-source slouží jako SMTP generátor a je součástí doručovacího systému Postfix. Díky tomuto nástroji je možné vygenerovat potřebnou zátěž na poštovním serveru. V průběhu testování jsem zaznamenával čas, v jakém byly zprávy odeslány, a průměrné zatížení CPU poštovního serveru, kde byly vygenerované zprávy doručovány. Pro testování byla zvolena konstantní velikost odesílaného těla emailu, která činila 10 kilobyte. Kvůli prokazatelnějším výsledkům bylo každé měření 5x opakováno a z těchto výsledků byla následně vypočítána jejich střední hodnota.

Testování poštovního serveru probíhalo následujícím způsobem:

time smtp-source –s <10/20/40/60/80/100> –l 10000 –m <100/1000> –f test1@mojedomena.cz –t testn@mojedomena.cz <IP/Hostname>

- kde s <10/20/40/60/80/100> udává počet paralelních požadavků

- kde l 10000 udává velikost těla zprávy
- kde m <100/1000> udává počet zaslaných zpráv
- kde f udává hlavičku odesílatele
- kde t udává, komu bude pošta doručena
- kde <IP/Hostname> odkazuje na poštovní server

Zhodnocení naměřených výsledků

Měření probíhalo ve dvou etapách. V první etapě byla nastavena hodnota počtu zaslaných zpráv na 100 a ve druhé etapě byla nastavena na hodnotu 1000. Při měření, kde byl nastaven počet zasílaných zpráv na 1000 a počet souběžných požadavků se blížil k 80, již poštovní server nedokázal takový provoz zvládat a hlásil chybu *Temporary table lookup failure*, proto je měření realizováno pouze do 60 souběžných požadavků.

V následujících tabulkách je patrné, že při použití nativního řešení je opět dosaženo lepších výsledků než u služby, která je virtualizována. Doba průběhu testu, jak u virtualizované, tak nativní služby, nedosahovala až tak velkých časových rozdílů, avšak využití procesorové jednotky CPU mluví o opaku.

Tabulka 6 - Naměřené hodnoty poštovního serveru – nativní služba, počet zaslaných zpráv 100

Počet souběžných požadavků	Doba průběhu testu [s]	Vytížení CPU [%]
10	0,816	16
20	0,626	15,4
40	0,583	25,8
60	0,586	24,8
80	0,562	28
100	0,544	26,4

Tabulka 7 - Naměřené hodnoty poštovního serveru – virtualizovaná služba, počet zaslaných zpráv 100

Počet souběžných požadavků	Doba průběhu testu [s]	Vytížení CPU [%]
10	0,739	51
20	0,768	62,4
40	0,754	73,2
60	0,775	58,4
80	0,744	83,4
100	0,728	75,6

Tabulka 8 - Naměřené hodnoty poštovního serveru – nativní služba, počet zaslaných zpráv 1000

Počet souběžných požadavků	Doba průběhu testu [s]	Vytížení CPU [%]
10	13,686	18
20	10,359	26,2
40	8,558	27,4
60	7,634	41,4

Tabulka 9 - Naměřené hodnoty poštovního serveru – virt. služba, počet zaslaných zpráv 1000

Počet souběžných požadavků	Doba průběhu testu [s]	Vytížení CPU [%]
10	14,606	96
20	10,982	98,4
40	8,778	100
60	6,809	100

5 Srovnávací testy HW řešení

V této části diplomové práce jsem se zaměřil na výkonové porovnání RouterOS x86 oproti dostupným HW řešením Cisco router 2811 a Cisco ASA 5510. Zaměřil jsem se na 3 základní veličiny, a to na testování propustnosti, testování zpoždění a v poslední řadě na testování ztrátovost. V první části jsem testované aktivní prvky stručně popsal a následně provedl porovnání, co se týče jejich vlastností jejich operačních systémů.

5.1 Cisco Router 2811

Jedná se o hardwarový směrovač vycházející ze série 2800. Tato série je určena spíše pro středně velké firmy, kde se směrovač využívá jako přístupový respektive hraniční. Operační systém, který je provozován na tomto směrovači, se nazývá Cisco IOS (Internet Operating System). Tento operační systém obsahuje nepřeberné množství funkcí, příkladem mohou být funkce pro směrování, jedná se o podporu dynamicky směrovaných protokolů (OSPF, BGP, IS-IS), dále se může jednat o bezpečnostní funkce, hlasové funkce apod. Každý směrovač vycházející ze série 2800 je možné osadit rozšiřujícími kartami, a navýšit tak počet vstupních portů. [11]

Cisco router 2811 disponuje těmito fyzickými vlastnostmi:

- Verze operačního systému IOS 12.4 (24)T1
- Operační paměť o velikosti 256 MB DDR
- 2 porty 100Base-T
- 2 porty pro USB
- 1 konzolový port

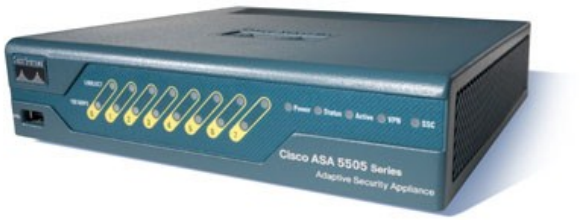
Obrázek 14 - Cisco router 2811 [12]

5.2 Cisco ASA 5510

Cisco ASA (Adaptive Security Appliances) je hardwarový firewall, který vychází z řady 5500 a je zaměřen především na zabezpečení služeb pro malé a střední podniky. Cisco ASA má v sobě zahrnut vysoce robustní firewall, IPS (Intrusion Prevention System) jedná se o systém umožňující detekovat pokusy o neautorizovaný přístup a tyto útoky následně zastavit, dále umožňuje realizovat zabezpečené sítě typu SSL VPN (Virtual Private Network) u realizace virtuálních sítí je nutné zakoupení patřičného počtu licencí. [13]

Cisco ASA 5510 disponuje těmito fyzickými vlastnostmi:

- Verze operačního systému ASA 7.0 (6)
- Operační paměť o velikosti 1 GB
- 5 portů 100Base-T
- 2 porty 1000Base-T
- 2 porty USB
- 1 port pro konzoli



Obrázek 15 - Cisco ASA 5510 [14]

5.3 Porovnání operačních systému RouterOS a Cisco IOS / ASA

V následující části práce jsem se pokusil o obecné porovnání operačních systémů Mikrotik RouterOS a Cisco IOS / ASA. Jedná se o hojně využívané operační systémy, které jsou nasazovány v přenosových sítích. Porovnání těchto operačních systémů jsem pojal jako vypsání výhod a nevýhod pro daný operační systém.

5.3.1 Mikrotik RouterOS

Výhody:

- Jednoduchá a pohodlná konfigurace v grafickém prostředí winbox či webfig
- Cenová dostupnost
- Možnost provozování na různých hardwarových platformách (x86, powerpc, apod.)
- Možnost provozování virtuálních strojů
- Pokročilé funkce směrování, firewallu

Nevýhody:

- Doprovázení častých chyb v implementaci
- Pomalá implementace nových technologií
- Uzavřený zdrojový kód

5.3.2 Cisco IOS / ASA

Výhody:

- Ověřený stabilní systém
- Možnost pohodlné konfigurace pomocí webového grafického rozhraní ASDM
- Podpora nejnovějších síťových technologií
- Kvalitní podpora, možnost certifikovaných školení (CCNA, CCNP, apod.)
- Implementován systém pro prevenci průniku - IPS
- Implementována technologie NAC (Network Access Control)

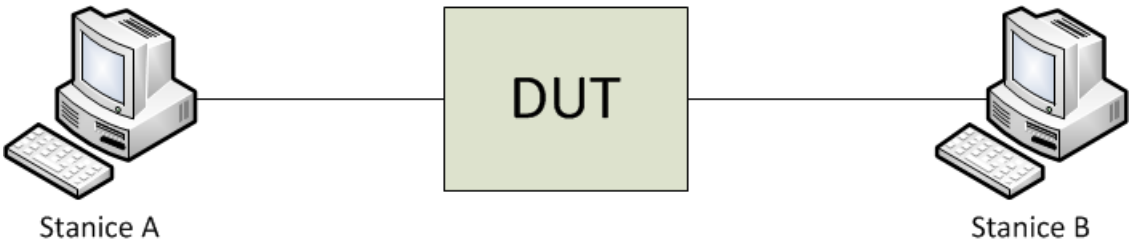
Nevýhody:

- Vysoká cena
- Nutnost zakoupení licencí pro specifické funkce (SSL VPN)
- Nemožnost provozování virtuálních strojů
- Vázáno na pořízený hardware

5.4 Metodika testování dle RFC

V této části kapitoly se budu zabývat možnými způsoby měření vlastností aktivních síťových prvků. Budu se snažit opírat o doporučené metody, které jsou popsány ve standardech RFC 2544 a RFC 1242. Tyto dokumenty obsahují pouze teoretické postupy, jak dosáhnout cíle, nikoliv však technické prostředky. Dokument RFC 2544 popisuje, jak nejvhodněji zapojit testované síťové zařízení, které je zde uváděno, jako DUT (Device Under Test) a je na něj nahlíženo jako na tzv. „černou skříňku" – tedy zařízení s jasně definovaným vstupně-výstupním rozhraním, avšak vnitřní struktura zůstává skryta.

Schéma realizace měření znázorněné na obrázku číslo 6 využívá dvě testovací stanice – stanice A slouží k odesílání dat, stanice B data přijímá, přičemž mezi stanicemi je nutné provádět synchronizaci stavových dat. [15]



Obrázek 16 - Schéma měření DUT

V další části dokumentu RFC 2544 je dále doporučeno každý test měření provádět několikrát, avšak nejméně pro 5 různých velikostí rámců. Je doporučeno provádět měření pro standardní délky Ethernetových rámců a to pro 64, 128, 256, 512, 1024, 1280 a 1518 bytů. [15]

5.4.1 Měření propustnosti (Throughput)

Testování propustnosti si klade za cíl zjistit maximální rychlost zasílání rámců, při které není žádný zahozen. Tato veličina je udávána v počtu přenesených bitů nebo rámců za jednotku sekundy.

Postup: Pošleme určitý počet rámců o určité rychlosti skrze testované zařízení. Poté sečteme rámce, které byly přeneseny testovaným zařízením. Pokud se počet přenesených rámců rovná počtu přijatých rámců, zvýšíme rychlost odesílání rámců a pokračujeme až do té doby, než jsou některé rámce zahozeny. [15]

5.4.2 Měření zpoždění (Latency)

Zpoždění je chápáno jako doba, po kterou byl jeden rámec přenášen přes testované zařízení. Rozlišují se dva typy zpoždění, a to zpoždění jednosměrné a obousměrné. Jednosměrné zpoždění je

čas, který uplyne mezi odesláním paketu zdrojem a jeho přijetím v cíli. Obousměrné zpoždění představuje čas cesty rámce od zdroje k cíli a zpět, plus i čas potřebný pro zpracování v cíli. Obousměrné zpoždění se též nazývá jako RTT (Round-Trip time) a používá se nejčastěji, jelikož je lépe měřitelné z jednoho místa. Cílem měření je získat velikost obousměrného zpoždění.

Postup: V první řadě je nutné nejdříve určit propustnost testovaného zařízení pro každý z uvedených velikostí rámců. Poté se pošle proud rámců s určitou velikostí skrze testované zařízení. Proud rámců by měl být zasílán nejméně po dobu 120 sekund a měl by obsahovat jeden specifický rámec vyslaný po 60 sekundách. V okamžiku, kdy je tento specifický rámec vyslán, je zaznamenán čas. Přijímací stanice musí tento rámec rozpoznat a zaznamenat čas, kdy byl rámec přijat. Zpoždění je pak vypočítáno jakou rozdíl zaznamenaných časů. Tento test je doporučeno opakovat nejméně dvacetkrát z důvodu přesnosti měření. [15]

5.4.3 Měření ztrátovosti (Frame Loss Rate)

Ztrátovost představuje procentuální údaj, vyjadřující kolik rámců mělo být přeneseno testovaným zařízením, ale z nedostatku systémových prostředků přeneseno nebylo.

Postup: Postup je obdobný jako u měření propustnosti. Vyšleme určitý počet rámců s určitou rychlostí, které by mělo testované zařízení přeposlat dále. [15]

Ztrátovost vypočteme dle vztahu:

$$Z = \frac{((A - B) * 100)}{A}$$

kde A představuje počet vstupních rámců a B počet výstupních rámců. Při prvním pokusu měření je zvolena rychlost odesílání, která odpovídá 100% maximální propustnosti pro danou velikost rámců. Další pokusy jsou spuštěny vždy o 10 % menší odesílací rychlostí, dokud se nenaleznou dva po sobě jdoucí pokusy, kdy se neztratí žádný rámec.

5.5 Použité měřicí přístroje

Zvolit vhodný měřící nástroj, který by plně podporoval doporučení dle RFC 2544, bylo velmi obtížné. Vyzkoušel jsem celou řadu softwarových nástrojů, jako jsou: iperf, netperf apod., ale ani jeden nedisponoval požadavky na něj kladenými. Nakonec se mi naskytla možnost využít hardwarového testeru od firmy Fetest, přesněji LAN tester ParaScope GigE. Tento síťový tester vyniká nepřeberným množstvím funkcí a umožňuje snadné a rychlé měření, dle doporučení RFC 2544.

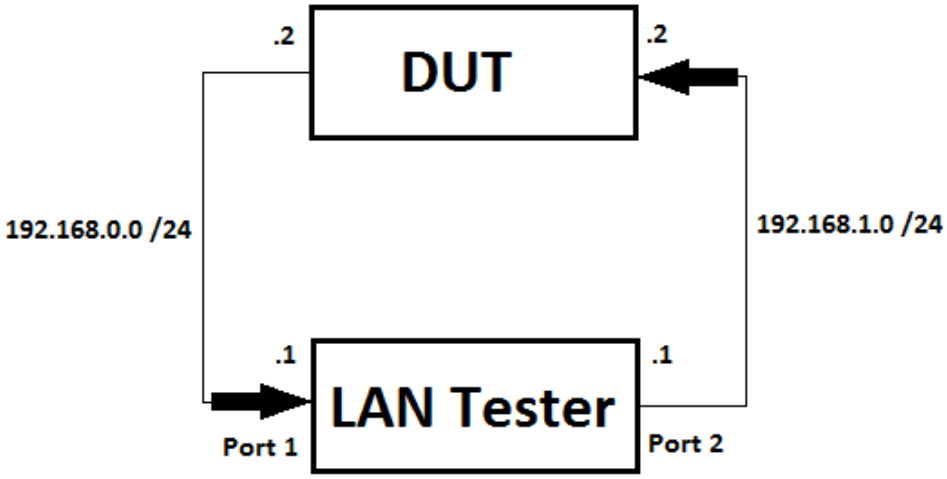
Základní vlastnosti přístroje:

- Dva metalické porty 10/100/1000Base-T a dva optické 100/1000Base-X
- Testování dle doporučení RFC 2544
- Podpora VLAN, VLAN Q-in-Q
- Měření a vyhodnocení parametrů na L2 a L3, možnost nastavení filtrů
- Export výsledků přes integrovaný USB port

Obrázek 17 - LAN tester ParaScope GigE

5.6 Realizace měření

Jelikož některá zařízení nedisponovala rychlejšími síťovými porty než 100Base-T, bylo nutné veškerá měření realizovat na této přenosové úrovni z důvodu stejného porovnání. Měření bylo realizováno dle schématu, které je znázorněno na obrázku číslo 8. Na testovaných zařízeních byla nakonfigurována pouze síťová rozhraní a následně směrování mezi nimi.



Obrázek 18 - schéma zapojení

5.6.1 Měření propustnosti

Z naměřených výsledků mohu konstatovat, že síťová propustnost u všech testovaných aktivních prvků dosahovala víceméně přibližných hodnot. Hlavní rozdíly byly zaznamenány při použití malých velikostí rámců a to 64 a 128bytů.

Směrovač na bázi RouterOS x86 dopadl v tomto testu nejhůře. Jeho přenosová rychlost u zvoleného rámce o velikosti 64bytů dosahovala pouze 18,7Mbps. Pokud je potřeba přenášet data v síti s vysokou rychlostí, není u něj vhodné volit tuto velikost rámce.

Nejlepšími výsledky disponoval směrovač Cisco router 2811, kde i při zvolení malých rámců dosahoval propustnosti od 76,1 až 100Mbps. Propustnost u aktivního prvku Cisco ASA se pohybovala mezi 31,2 až 100Mbps.

Tabulka 10 - Měření propustnosti RouterOS x86

Velikost rámce	Teoretická rychlost	Naměřená rychlost	Naměřená rychlost
[B]	[FPS]	[FPS]	[Mbps]
64	148809	27829	18,7
128	84459	68577	81,2
256	45289	45290	100,0
512	23496	23496	100,0
1024	11973	11973	100,0
1280	9615	9615	100,0
1518	8127	7616	93,7

Tabulka 11 - Měření propustnosti Cisco Router / Cisco ASA

	Cisco router 2811		Cisco ASA 5510	
Velikost rámce	**Naměřená rychlost**		**Naměřená rychlost**	
[B]	[FPS]	[Mbps]	[FPS]	[Mbps]
64	148802	76,1	46460	31.2
128	82769	84,7	47467	56.2
256	45290	100.0	45290	100.0
512	23496	100.0	23496	100.0
1024	11973	100.0	11973	100.0
1280	9615	100.0	9615	100.0
1518	8127	100.0	8127	100.0

5.6.2 Měření zpoždění a ztrátovosti

Co se týče naměřených výsledků zpoždění opět dle testu dosahoval nejlepších hodnot Cisco router 2811, který dosahoval 148µs při velikosti rámce 1518bytů, za ním se umístila Cisco ASA 5510 a v závěru RouterOS x86, jehož zpoždění při stejné velikosti rámce jako u Cisco router dosahovalo 248µs. Zpoždění má velký vliv na služby pracující v reálném čase, pokud jsou hodnoty příliš vysoké, úměrně se podepisují na degradaci kvality přenosu hlasu, videa a dalších služeb.

Co se týče měření ztrátovosti, dopadly testované zařízení následovně. Nejlepších hodnot dosahoval Cisco router 2811, který měl ztrátovost pouze 0,1% při použití rámce 64 bytů, za ním se umístil směrovač na bázi RouterOS x86, jehož ztrátovost byla 10% při stejné délce rámce, a to 64bytů, nejhůře dopadla Cisco ASA 5510, která dosahovala ztrátovosti 66,2% při délce rámce 64bytů a 41,1% při délce rámce 128bytů. Ztrátovost rámců se negativně projevuje na zkreslení nebo také šumu při přenosu videa či hlasu.

Tabulka 12 - Měření zpoždění a ztrátovost - RouterOS x86

Velikost rámce	Zpoždění	Ztrátovost
[B]	[µs]	[%]
64	79	10,0
128	72	0,0
256	113	0,0
512	166	0,0
1024	214	0,0
1280	237	0,0
1518	248	0,1

Tabulka 13 - Měření zpoždění a ztrátovosti - Cisco Router / Cisco ASA

	Cisco router 2811		Cisco ASA 5510	
Velikost rámce	**Zpoždění**	**Ztrátovost**	**Zpoždění**	**Ztrátovost**
[B]	[µs]	[%]	[µs]	[%]
64	35	0,1	37	66,2
128	40	0,0	43	41,1
256	69	0,0	71	0,0
512	102	0,0	108	0,0
1024	139	0,0	145	0,0
1280	146	0,0	163	0,0
1518	148	0,0	163	0,0

Závěr

V této diplomové práci jsem se věnoval realizaci hraničního směrovače na bázi systému RouterOS x86 s funkcemi firewall, mail server a http server. Téma práce jsem pojal ve čtyřech rovinách, které jsem se při psaní práce případnému čtenáři snažil co nejvíce objasnit.

V části teoretické jsem zprvu popisoval operační systém RouterOS, jeho vlastnosti, možnosti nasazení, licencování. Zaměřoval jsem se hlavně na funkce, které jsou primárně využity v praktické části diplomové práce. Dále jsem zpracovával vlastní koncepci návrhu řešení hraničního směrovače, zaměřoval jsem se na výběr virtualizovaného systému, výběr a popis vhodných nástrojů určených pro realizaci poštovního a webového serveru. Ve finální teoretické části jsem se soustředil na návrh metodiky pro srovnání řešení založených na nativním použití služby, použití RouterOS a na HW řešení.

V části praktické byla otestována jediná dostupná virtualizační metoda určená pro hardwarovou platformu x86, KVM. Na této virtualizační metodě byla hostována Linuxová distribuce Ubuntu, kde byly následně nakonfigurovány a zprovozněny služby řešící poštovní a webový server. Pro realizaci poštovního serveru byl vybrán a následně nakonfigurován Postfix, který byl v roli SMTP serveru, a Dovecot v roli POP3/IMAP serveru. Webový server pak tvořila trojice Apache, MySQL a PHP.

Při instalaci hostovaného systému jsem nezaznamenal žádné problémy, které by neumožňovaly nasazení linuxové distribuce Ubuntu, instalace i samotné konfigurace služeb proběhly bez sebemenších problémů.

Dále jsem v praktické části provedl několik testů, přičemž jsem se snažil o porovnání chování služeb, které jsou v ROS virtualizovány, oproti službám nativním. Z dosažených výsledků je patrné, že nasazené služby, které jsou řešeny pomocí virtualizace, za službami nativními značně ztrácejí. U testování webového serveru byly požadavky vyřízeny v daleko kratší době než u služby virtualizované. U testování poštovního serveru nebyl časový rozdíl mezi službou nativní a virtualizovanou pro zpracování požadavků tak rozdílný. Jediným rozdílem bylo vytížení procesorové jednotky CPU, kde bylo dosaženo vytížení až na hodnotu 100 % u virtualizované služby, u služby nativní bylo maximální vytížení CPU na 41,4 %.

Na základě poznatků z měření nedoporučuji používat virtualizované služby v ROS v sítích, kde se pohybuje velké množství přenášených dat. Případné použití vidím v nasazení do velmi malých sítí, kde na výkon směrovače není brán takový zřetel.

V další praktické části jsem předvedl základní konfiguraci pro zabezpečení hraničního směrovače pomocí stavového firewallu.

V poslední praktické části jsem se zaměřil na porovnání RouterOS x86 oproti dostupným hardwarovým řešením Cisco router 2811 a Cisco ASA 5510. Cílem měření bylo porovnat výkonnostní rozdíly parametrů aktivních prvků, jejichž architektura je řešena rozdílnými způsoby. Měření bylo realizováno pomocí zapůjčeného hardwarového generátoru. Zaměřil jsem se na 3 základní veličiny, a to na testování propustnosti, testování zpoždění a testování ztrátovosti. Z dosažených výsledků z měření jsem vypozoroval, že při zvolení malých velikostí rámců a to 64bytů a 128bytů dosahoval RouterOS x86 menších přenosových rychlostí než konkurenti firmy Cisco. Co se týče testování, zpoždění RouterOS x86 dosahoval 248µs při zvolení rámce 1518bytů, Cisco router při stejné velikosti rámce 148µs a Cisco ASA dosahovala zpoždění 163µs. Co se týče ztrátovosti, nejlepších hodnot dosahoval Cisco router, který měl ztrátovost pouze 0,1 % při použití rámce o velikosti 64bytů. RouterOS x86 dosahoval ztrátovosti 10 % při téže velikosti rámce. Cisco ASA dosahovala ztrátovosti 66% při velikosti rámce 64bytů a 41,1 % při velikosti rámce 128bytů.

Z celkových dosažených výsledků v porovnání Cisco router a Cisco ASA nedopadl RouterOS x86 na mnou konfigurované PC sestavě nikterak špatně.

Hlavním cílem této diplomové práce byla realizace hraničního směrovače na bázi systému RouterOS x86 s funkcemi firewall, mail server a http server, což se mi podařilo splnit, dále jsem provedl měření na základě, kterých jsem porovnal řešení založené na nativním použití služby, použití RouterOS a na HW řešení.

Doufám, že má práce poskytne případným čtenářům informace, které hledají.

Literatura

[1] ŠTRAUCH, Adam. Mikrotik: seznámení s Wi-Fi krabičkou. *Mikrotik: seznámení s Wi-Fi krabičkou* [online]. 2008 [cit. 2013-07-31]. Dostupné z: http://www.root.cz/clanky/mikrotik-seznameni-s-wi-fi-krabickou/

[2] Manual:TOC. *Manual:TOC* [online]. 2013 [cit. 2013-07-31]. Dostupné z: http://wiki.mikrotik.com/wiki/Manual:TOC

[3] Vítejte v Debianu. *Vítejte v Debianu* [online]. [cit. 2013-07-31]. Dostupné z: http://www.debian.org/releases/stable/s390x/ch01s02.html.cs

[4] Dokumentace k Ubuntu: Oficiální dokumentace. *Dokumentace k Ubuntu: Oficiální dokumentace* [online]. [cit. 2013-07-31]. Dostupné z: http://www.ubuntu.cz/podpora/dokumentace

[5] DOČEKAL, Michal. Správa linuxového serveru: Instalace LAMP. *Správa linuxového serveru: Instalace LAMP* [online]. 2010 [cit. 2013-07-31]. Dostupné z: http://www.linuxexpres.cz/praxe/sprava-linuxoveho-serveru-instalace-lamp

[6] ZAJÍC, Petr. MySQL: Pestrý svět databází. *MySQL* [online]. 2005 [cit. 2013-07-31]. Dostupné z: http://www.linuxsoft.cz/article.php?id_article=731

[7] PHP: Základy PHP. [online]. [cit. 2013-07-31]. Dostupné z: http://www.tvorba-webu.cz/php/

[8] MySQL : L'outil PHPMyAdmin. [online]. 2012 [cit. 2013-07-31]. Dostupné z: http://www.php-astux.info/mysql-phpmyadmin.php

[9] PETERKA, Jiří. Architektura elektronické pošty. *Architektura elektronické pošty* [online]. [cit. 2013-07-31]. Dostupné z: http://www.earchiv.cz/a98/a807k180.php3

[10] JELÍNEK, Lukáš. Stavíme poštovní server. [online]. [cit. 2013-07-31]. Dostupné z: http://www.abclinuxu.cz/serialy/stavime-postovni-server

[11] CISCO. *Cisco 2811 Integrated Services Router: Cisco router* [online]. [cit. 2013-07-31]. Dostupné z: http://www.cisco.com/en/US/products/ps5881/index.html

[12] CISCO. *Cisco Router 2811* [online]. [cit. 2013-07-31]. Dostupné z: http://img1.tradeget.com/hkyejian03%5CHEA9AN161cisco_2811.jpg

[13] CISCO. *Cisco ASA 5500: Cisco ASA* [online]. [cit. 2013-07-31]. Dostupné z: http://www.cisco.com/en/US/products/ps6120/products_data_sheets_list.html

[14] CISCO. *Cisco ASA 5510: Cisco ASA* [online]. [cit. 2013-07-31]. Dostupné z: http://itknowledgeexchange.techtarget.com/cisco/sadly-the-pix-firewall-is-discontinued/

[15] Benchmarking Methodology for Network Interconnect Devices: Document RFC. HARVARD UNIVERSITY. [online]. 1999 [cit. 2013-07-31]. Dostupné z: http://www.ietf.org/rfc/rfc2544.txt

Seznam obrázků

Seznam tabulek

Seznam příloh

Příloha 1 – přiložené CD s obsahem:

- Diplomova_prace
- Konfigurace
    - Konfigurace_Dovecot
    - Konfigurace_Postfix
    - Konfigurace_PostfixAdmin
    - Konfigurace_RouterOS
    - Konfigurace_RouterOS
- Mereni_sluzeb
    - Mereni_HW_dle_RFC
    - Nativni_mereni_Postovniho_serveru
    - Nativni_mereni_Weboveho_serveru
    - Virtualizovane_mereni_Postovniho_serveru
    - Virtualizovane_mereni_Weboveho_serveru
- Overeni_funkcnosti_sluzeb_SMTP,POP3,IMAP